

**PURWARUPA SISTEM OTOMASI PERAWATAN
TANAMAN CABAI PADA SMART GREENBOX
BERBASIS IOT**

**(*PROTOTYPE OF CHILLI PLANTS AUTOMATION
SYSTEM IN IOT-BASED SMART GREENBOX*)**

TUGAS AKHIR

Disusun dalam rangka memenuhi salah satu persyaratan untuk menyelesaikan
Mata Kuliah Tugas Akhir S1 Teknik Telekomunikasi

Oleh

Ikram Andika Ukar

1101181299



Telkom
University

**FAKULTAS TEKNIK ELEKTRO
UNIVERSITAS TELKOM
BANDUNG
2022**

	UNIVERSITAS TELKOM	No. Dokumen	
	Jl. Telekomunikasi No.1 Ters. Buah Batu Bandung 40257	No. Revisi	
	FORMULIR LEMBAR PENGESAHAN	Berlaku Efektif	

LEMBAR PENGESAHAN

TUGAS AKHIR

PURWARUPA SISTEM OTOMASI PERAWATAN TANAMAN CABAI PADA SMART GREENBOX BERBASIS IOT

***PROTOTYPE OF CHILLI PLANTS AUTOMATION SYSTEM IN IOT-
BASED SMART GREENBOX***

Telah disetujui dan disahkan sebagai buku tugas akhir

Program Studi S1 Teknik Telekomunikasi

Fakultas Teknik Elektro

Universitas Telkom

Disusun Oleh:

IKRAM ANDIKA UKAR

1101181299

Bandung, 19 Februari 2022

Disahkan Oleh:

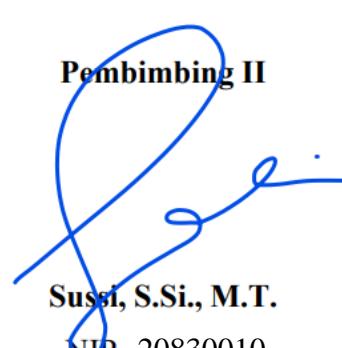
Pembimbing I



DR. Nyoman Bogi Aditya Karna, S.T., MSEE.

NIP. 11730049

Pembimbing II



Sussi, S.Si., M.T.

NIP. 20830010

	UNIVERSITAS TELKOM	No. Dokumen	
	Jl. Telekomunikasi No.1 Ters. Buah Batu Bandung 40257	No. Revisi	
	FORMULIR LEMBAR PERNYATAAN ORISINALITAS	Berlaku Efektif	

LEMBAR PERNYATAAN ORISINALITAS

Nama : Ikram Andika Ukar
 NIM : 1101184009
 Alamat : Gerlong Residence no. A9, Gegerkalong Hilir, Kota Bandung
 No, Telepon : 087821637040
 Email : ikramau17@gmail.com

Menyatakan bahwa Tugas Akhir ini merupakan karya orisinal sendiri, dengan judul:

**PURWARUPA SISTEM OTOMASI PERAWATAN TANAMAN CABAI
PADA SMART GREENBOX BERBASIS IOT**

***PROTOTYPE OF CHILLI PLANTS AUTOMATION SYSTEM IN IOT-
BASED SMART GREENBOX***

Atas pernyataan ini, saya siap menanggung sanksi yang dijatuhkan apabila kemudian ditemukan adanya pelanggaran terhadap kejujuran akademik atau etika keilmuan dalam karya ini, atau ditemukan bukti yang menunjukkan ketidakaslian karya ini.

Bandung, 19 Februari 2022



IKRAM ANDIKA UKAR

NIM. 1101181299

ABSTRAK

Pada zaman ini, banyak orang yang memanfaatkan teknologi untuk menjadi solusi dalam permasalahan dan pekerjaan yang dihadapi oleh manusia. Contohnya adalah hobi manusia pada zaman ini di perkotaan adalah bercocok tanam di rumah. Masalah yang dihadapinya adalah mayoritas masyarakat di perkotaan itu sibuk dengan pekerjaannya dari pagi hingga sore hari sehingga tanaman yang ia rawat itu sulit untuk dipantau dan dikontrol. Kehadiran teknologi *Internet of Things* (IoT) itu menjadi gebrakan besar untuk permasalahan-permasalahan yang ada di masyarakat. Dengan IoT, masyarakat bisa dipermudah untuk memantau dan mengontrol tanamannya setiap saat dari jarak jauh.

Robot adalah sebuah sistem berupa alat yang dapat bergerak sesuai apa yang diprogram dengan komputer atau kodingan. Robot otomasi yang diterapkan pada tanaman cabai di penelitian ini adalah robot kartesius dengan konsep mesin CNC. Robot jenis ini biasa digunakan untuk otomasi di pabrik-pabrik. Lengan dari robot ini berupa alat penyiram dan pemupuk tanaman yang dapat berpindah sesuai koordinat tanaman yang tercatat.

Penelitian ini melakukan pembangunan purwarupa sistem otomasi berbasis IoT terhadap tanaman cabai yang berada dalam greenbox. Desain purwarupa sistem otomasi penelitian ini adalah berupa robot CNC kartesian dengan sumbu X dan Y. Purwarupa melakukan otomasi tergantung pada input nilai dari kelembaban tanah atau pH tanah dari sensor dengan transmisi data menggunakan LoRa. Metode pengujian performansi alat berupa pengujian jarak, akurasi, kemampuan pengulangan, waktu proses otomasi, dan kualitas transmisi data.

Hasil dari penelitian ini adalah tercapainya sistem otomasi penyiraman atau pemberian pupuk cair terhadap greenbox yang berisi tanaman cabai. Otomasinya adalah penyiraman atau pemberian pupuk cair berdasarkan input dari hasil sensor. Tingkat akurasi alat yang didapatkan mencapai 95,53% untuk sumbu X dan 99,15% untuk sumbu Y. Penerapan purwarupa sistem otomasi ini tergolong efisien dalam penggunaan tenaga manusia. Kualitas pengiriman data yang diterima oleh LoRa juga tergolong baik.

Kata kunci : IoT, robot otomasi, penyiraman, kelembaban tanah, pH tanah, CNC, robot kartesius, pertumbuhan tanaman.

ABSTRACT

In this day and age, many people use technology to be a solution to the problems and jobs faced by humans. An example is the human hobby of today in urban areas is farming at home. The problem he faces is that the majority of people in urban areas are busy with their work from morning to evening, so the plants he cares for are difficult to monitor and control. The presence of Internet of Things (IoT) technology is a big breakthrough for the problems that exist in society. With IoT, it is easier for people to monitor and control their plants at any time remotely.

Robot is a system in the form of a tool that can move according to what is programmed with a computer or code. The automation robot applied to chili plants in this study is a Cartesian robot with the concept of a CNC machine. This type of robot is commonly used for automation in factories. The arm of this robot is a sprinkler and plant fertilizer that can move according to the coordinates of the recorded plants.

This study carried out the development of a prototype of an IoT-based automation system for chili plants in a greenbox. The prototype design of this research automation system is a CNC Cartesian robot with X and Y axes. The prototype performs automation depending on the input value of soil moisture or soil pH from the sensor with data transmission using LoRa. The method of testing the performance of the tool is in the form of testing distance, accuracy, repeatability, automation process time, and data transmission quality.

The result of this research is the achievement of an automation system for watering or giving liquid fertilizer to the greenbox containing chili plants. The automation is watering or giving liquid fertilizer based on input from sensor results. The accuracy of the obtained tools reaches 95.53% for the X axis and 99.15% for the Y axis. The application of this prototype automation system is considered efficient in the use of human power. The quality of data transmission received by LoRa is also quite good.

Keywords : IoT, automation robot, watering, soil moisture, soil pH, CNC, cartesian robot, plants growth.

KATA PENGANTAR



Assalamu 'alaikum Wr. Wb.,

Alhamdulillahirabbilalamin, pertama dan yang paling utama, segala puji dan syukur penulis panjatkan kehadirat Allah SWT atas segala nikmat dan karunia-Nya, sehingga penulis dapat menyelesaikan tugas akhir dengan judul

PURWARUPA SISTEM OTOMASI PERAWATAN TANAMAN CABAI PADA SMART GREENBOX BERBASIS IOT

Penulis telah melaksanakan serangkaian penelitian yang pada akhirnya ditulis kedalam buku tugas akhir ini dengan baik sebagai prasyarat menyelesaikan pendidikan sarjana program studi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom.

Penulis menyadari masih banyak kekurangan dalam penyusunan tugas akhir ini yang disebabkan oleh keterbatasan penulis. Oleh karena itu, dengan kerendahan hati penulis mohon maaf sebesar-besarnya dan besar harapan penulis mendapatkan kritik dan saran yang membangun.

Akhir kalam besar harapan penulis agar tugas akhir ini bisa menjadi manfaat untuk semua pihak, serta dapat dikembangkan menjadi sebuah penelitian yang lebih terbarukan.

Wassalamu 'alaikum Wr. Wb.

Bandung, 19 Februari 2022

Ikram Andika Ukar

UCAPAN TERIMA KASIH

Tugas akhir merupakan perjalanan akhir dari penulis sebagai mahasiswa Program Studi S1 Teknik Telekomunikasi yang telah ditempuh selama 3,5 tahun ini. Dalam penulisan tugas akhir ini dan juga selama masa studi, penulis telah menghadapi banyak sekali hambatan dan rintangan. Namun, semua hambatan dan rintangan telah penulis lalui atas berkat banyak pihak yang telah membantu, membimbing, dan mendukung penulis selama ini. Oleh karena itu, penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Allah S.W.T, atas segala nikmat dan rezeki untuk seluruh makhluk hidup di dunia ini.
2. Rasulullah S.A.W, atas pedoman untuk hidup yang sangat bermanfaat melalui mukjizat terdahsyat, yaitu Al-Qur'an dan segala hadits.
3. Karyadwina Ukar dan Anne Dewi Kania, selaku orang tua penulis yang senantiasa merawat penulis dari kecil hingga saat ini dengan penuh kasih sayang dan kesabaran.
4. Ayyub Nasrah Atmadja dan Nurul Azizah, selaku teman seperjuangan dalam mengerjakan tugas akhir ini.
5. Dr. Nyoman Bogi Aditya Karna, S.T., MSEE., selaku pembimbing 1 yang selalu memberi dukungan dan saran yang sangat membantu.
6. Sussi, S.Si., M.T., selaku pembimbing 2 yang selalu semangat dalam membimbing penulis.
7. Cynthia Maulina Noor, atas kehadirannya di kehidupan penulis yang sangat berpengaruh dalam kenyamanan, motivasi, dan kebahagiaan penulis.
8. Dr. Koredianto Usman, S.T., M.Sc., selaku dosen wali penulis yang senantiasa menjawab permasalahan-permasalahan penulis.
9. Dr. Levy Olivia Nur, S.T., M.T., selaku Kepala Program Studi S1 Telco yang telah memimpin program studi ini dengan sangat baik.
10. Seluruh Dosen S1 Telco Tel-U, atas jasanya dalam menyampaikan ilmu yang bermanfaat kepada penulis dan mahasiswa lainnya dengan baik.
11. Keluarga besar penulis, atas dukungan dan doa yang telah diberikan.
12. Elegos & Asely, selaku sahabat-sahabat yang senantiasa saling mendukung, membantu, dan menghibur satu sama lain.
13. Kadal Gurun, selaku sahabat-sahabat yang mendorong dan membantu penulis untuk menjaga nilai akademik serta mengembangkan keterampilan.
14. Kuliner Bandung, selaku sahabat-sahabat yang selalu ada dan telah melewati banyak momen sekaligus rintangan yang berkesan.
15. Elite KPM, selaku sahabat-sahabat yang selalu memberi dukungan walaupun belum lama kenal dekat dengan penulis.
16. Kelas TT-42-04, atas segala kenangan selama 3,5 tahun ini yang sangat berkesan.

17. BP HMTT, sebagai wadah yang memberikan penulis banyak sekali pelajaran dan pengalaman melalui ASTERISK, TESLA, Staf Muda, Staf Ahli, BSO, Inti Kabinet, dan masih banyak lagi.
18. Angkatan Telco XLII, selaku angkatan terbaik bagi penulis yang saling bahu-membahu membangun masa depan secara bersama-sama.
19. Angkatan Telco XLI, XL, XXXIX, dan seterusnya, selaku angkatan senior yang senantiasa berbagi ilmu dan pengalaman yang bermanfaat.
20. Angkatan Telco XLIII dan XLIV, selaku angkatan junior yang senantiasa mendukung keberlangsungan organisasi dengan sepenuh hati.
21. Seluruh Staf Telkom University, atas pelayanan yang terbaik kepada seluruh mahasiswa.
22. Teman-teman BOT, UKM Riverside, UKM Tenis Lapangan, dan PKKMB, atas memori, kenangan, serta relasi yang sangat berkesan.
23. Seluruh teman baik penulis yang tidak bisa penulis sebut satu per satu, atas seluruh kebermanfaatan yang telah didapatkan oleh penulis.

Demikian ucapan terima kasih dari penulis yang tidak dapat penulis sebutkan satu-satu atas jasa luar biasa yang telah diberikan. Penulis juga ingin meminta maaf kepada seluruh pihak atas kesalahan yang telah disengaja ataupun tidak, serta budi yang belum terbalaskan oleh penulis. Semoga sukses untuk semua pihak yang terkait.

DAFTAR ISI

LEMBAR PENGESAHAN	ii
ABSTRAK	iv
ABSTRACT	v
KATA PENGANTAR	vi
UCAPAN TERIMA KASIH.....	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xiii
DAFTAR SINGKATAN	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan dan Manfaat.....	2
1.3 Rumusan Masalah	2
1.4 Batasan Masalah.....	2
1.5 Metode Penelitian.....	3
BAB II TINJAUAN PUSTAKA.....	4
2.1 <i>Internet of Things</i>	4
2.2 Tanaman Cabai Rawit	5
2.3 <i>Greenhouse In a Box / Greenbox</i>	7
2.4 Robot	8
2.4.1 Robot Koordinat Kartesius.....	8
2.4.2 <i>Computer Numerical Control (CNC)</i>	10
2.5 <i>Hardware</i>	10
2.5.1 <i>Microcontroller</i>	10
2.5.2 <i>Stepper Motor</i>	11
2.5.3 Arduino CNC Shield	14
2.5.4 LoRa (<i>Long Range</i>).....	15
2.5.5 Relay, Transistor, Resistor	16
2.6 <i>Software</i>	17
2.6.1 Arduino IDE.....	17
2.6.2 Fritzing	17

BAB III PERANCANGAN SISTEM	18
3.1 Konsep <i>Smart Greenbox</i>	18
3.1.1 Diagram Blok Sistem <i>Smart Greenbox</i>	18
3.1.2 Desain Sistem Purwarupa	19
3.1.3 Diagram Alir Kerja Sistem	19
3.2 Pembangunan Purwarupa	21
3.2.1 Alat dan Bahan Pembangunan Purwarupa.....	21
3.2.2 Tahapan Pembangunan Purwarupa	22
3.3 Desain Rangkaian Purwarupa	23
3.3.1 Desain Purwarupa	23
3.3.2 Rangkaian Purwarupa	26
3.4 Metode Analisis Performansi Alat	28
3.4.1 Pengujian Posisi	28
3.4.2 Pengujian Akurasi & <i>Repeatability</i>	29
3.4.3 Pengujian Waktu Proses Otomasi	30
3.5 Pengujian Parameter Pengiriman LoRa.....	32
3.5.1 RSSI (<i>Received Signal Strength Indicator</i>)	32
3.5.2 SNR (<i>Signal Noise Ratio</i>)	32
BAB IV HASIL KERJA SISTEM	34
4.1 Hasil Pembangunan Alat	34
4.2 Hasil Pengujian Posisi	37
4.3 Hasil Pengujian Akurasi dan <i>Repeatability</i>	38
4.4 Hasil Pengujian Waktu Proses Otomasi	40
4.5 Hasil Pengukuran Parameter Pengiriman LoRa	42
BAB V PENUTUP	45
5.1 Kesimpulan.....	45
5.2 Saran	45
DAFTAR PUSTAKA	46
LAMPIRAN	48

DAFTAR GAMBAR

Gambar 1: Konsep Dasar Umum IoT	5
Gambar 2: Cabai Rawit.....	6
Gambar 3: Tanaman Cabai Rawit	6
Gambar 4: Contoh Greenhouse Pada Umumnya	8
Gambar 5: Konsep Dasar Robot Kartesius 3 Sumbu	9
Gambar 6: Desain Umum 1 Robot Kartesius.....	9
Gambar 7: Desain Umum 2 Robot Kartesius.....	10
Gambar 8: Stepper Motor Nema 17	13
Gambar 9: Stepper Motor Driver DRV8825.....	13
Gambar 10: Denah Umum DRV8825.....	13
Gambar 11: Desain Umum Sistem Arduino Uno + DRV8825 + Stepper Motor .	14
Gambar 12: Arduino Uno CNC Shield V3	14
Gambar 13: Denah CNC Shield V3	15
Gambar 14: Relay	16
Gambar 15: Transistor.....	16
Gambar 16: Resistor.....	16
Gambar 17: Diagram Blok Sistem Smart Greenbox.....	18
Gambar 18: Desain Sistem Purwarupa Otomasi Penyiraman.....	19
Gambar 19: Diagram Alir Kerja Otomasi Penyiraman.....	20
Gambar 20: Tampak 1 Desain Purwarupa	24
Gambar 21: Tampak 2 Desain Purwarupa	24
Gambar 22: Tampak 3 Desain Purwarupa	25
Gambar 23: Tampak 4 Desain Purwarupa	25
Gambar 24: Tampak 5 Desain Purwarupa	26
Gambar 25: Rangkaian Skematik Arduino Uno - CNC Shield V3.....	26
Gambar 26: Rangkaian Komponen Sistem Purwarupa.....	27
Gambar 27: Diagram Alir Pengujian Posisi.....	29
Gambar 28: Diagram Alir Pengujian Akurasi dan Repeatability.....	30
Gambar 29: Diagram Alir Pengujian Waktu Proses Otomasi.....	31
Gambar 30: Rangkaian Arduino Uno + CNC Shield + LoRa di dalam Box X4 Custom	34
Gambar 31: Kerangka Besi dan Mesin CNC Purwarupa.....	34

Gambar 32: Penempatan Board dan Tanaman.....	35
Gambar 33: Tampak Depan Sistem Smart Greenbox	36
Gambar 34: Keseluruhan Sistem Smart Greenbox	37
Gambar 35: Data RSSI Pengiriman Data LoRa	44
Gambar 36: Data SNR Pengiriman Data LoRa.....	44

DAFTAR TABEL

Tabel 1: Spesifikasi Arduino Uno	11
Tabel 2: Spesifikasi Stepper Motor Nema 17	12
Tabel 3: Alat dan Bahan Pembangunan Purwarupa Otomasi Penyiraman	21
Tabel 4: Tata Hubung Pin Komponen LoRa dan Pompa Air	28
Tabel 5: Hasil Pengujian Posisi Sumbu X	37
Tabel 6: Hasil Pengujian Posisi Sumbu Y	38
Tabel 7: Hasil Pengujian Akurasi di Sumbu X	38
Tabel 8: Hasil Pengujian Akurasi di Sumbu Y	39
Tabel 9: Hasil Waktu Tempuh Pergerakan Nozzle ke Target Tanaman.....	41
Tabel 10: Hasill Waktu Tempuh Otomasi Penyiraman	42
Tabel 11: Kelebihan & Kekurangan Antar Algoritma.....	42
Tabel 12: Data Nilai RSSI dan SNR LoRa Receiver	43

DAFTAR SINGKATAN

IoT	<i>: Internet of Things</i>
RH	<i>: Relative Humidity</i>
LoRa	<i>: Long Range</i>
CNC	<i>: Computer Numerical Control</i>
IDE	<i>: Integrated Development Environment</i>
RSSI	<i>: Received Signal Strength Indicator</i>
SNR	<i>: Signal Noise Ratio</i>

BAB I

PENDAHULUAN

1.1 Latar Belakang

Teknologi di dunia ini selalu berkembang. Berkembang dalam arti dengan sendirinya maupun dikembangkan. Pengembangan teknologi ini ditujukan untuk mempermudah pekerjaan manusia serta menyelesaikan masalah dengan efektif dan efisien. Kita sekarang sedang memasuki Era Revolusi Industri 4.0 di mana salah satu pilar perkembangan industrinya adalah *Internet of Things* (IoT). IoT merupakan teknologi yang dapat membantu kita dalam hal *controlling* dan *monitoring* dari jarak jauh dengan bantuan internet[1]. Teknologi ini benar-benar dikembangkan pada zaman ini yang memungkinkan untuk teknologi ini menjadi makanan sehari-hari beberapa tahun yang akan datang.

Pada zaman ini, gaya hidup mayoritas orang di perkotaan sudah berbeda dengan puluhan tahun yang lalu. Banyak orang yang sibuk untuk bekerja di luar rumah seharian. Karena hal tersebut, banyak orang yang membutuhkan kegiatan yang dapat menyegarkan pikiran setelah bekerja seharian. Contohnya adalah dengan memelihara binatang, berolahraga, bercocok tanam, dan lain-lain. Namun, beberapa kegiatan itu perlu diperhatikan dan dirawat secara rutin sedangkan mayoritas orang tidak berada di rumah dari pagi sampai sore hari. Oleh karena itu, diperlukan sebuah teknologi yang dapat melakukan *controlling* dan *monitoring* dari jarak jauh, yaitu dengan IoT.

Bercocok tanam merupakan salah satu hobi yang cukup digemari oleh orang-orang di perkotaan pada saat ini. Bercocok tanam yang dimaksud adalah bercocok tanam di rumah sendiri. Beberapa orang menanam di pot secara langsung dan beberapa orang menggunakan *greenhouse*. Perangkat IoT yang diperlukan untuk *controlling* dan *monitoring* dapat dipasang di *greenhouse* juga[2]. Perangkatnya dapat berupa robot yang memiliki sensor beserta alat otomasi yang terhubung dengan internet sehingga data yang diambil dapat terunggah ke *database*. Dengan ini, masalah tanaman yang ditinggal di rumah seharian dapat teratasi.

Pertumbuhan tanaman memiliki beberapa faktor yang memengaruhi kecepatan dan kualitas dari pertumbuhan tersebut. Faktor-faktor tersebut pengaruhnya berbeda-beda tiap tanamannya. Faktor-faktor tersebut dapat dijadikan parameter atau variabel bebas dalam penelitian ini yang nantinya dapat dianalisis untuk menemukan parameter mana yang paling berpengaruh. Parameter-parameter yang dimaksud berupa intensitas cahaya, kelembaban udara, kelembaban tanah, dan pH tanah[3]. Jika nilai dari pH tanah yang didapatkan di bawah batas yang ditentukan (terlalu asam), maka akan dilakukan otomasi pengapuran dengan perangkat IoT. Penelitian ini diharapkan dapat membantu masyarakat yang berhalangan untuk memantau tanamannya setiap saat.

1.2 Tujuan dan Manfaat

Tujuan dari penelitian ini adalah sebagai berikut:

1. Membangun purwarupa perangkat otomasi IoT berupa alat robot kartesius sumbu X dan Y untuk menyiram otomatis.
2. Purwarupa dapat melakukan otomasi penyiraman atau pemberian pupuk cair otomatis ataupun terprogram terhadap tanaman yang ada di dalam *greenbox*.
3. Mendapatkan data hasil analisis performansi purwarupa.

Manfaat dari penelitian ini adalah sebagai berikut:

1. Menjadi solusi untuk melakukan *controlling* dan *monitoring* pertumbuhan tanaman dari jarak jauh.
2. Dapat dikembangkan dan direalisasikan untuk dipasarkan.

1.3 Rumusan Masalah

Berdasarkan latar belakang penelitian ini, terdapat beberapa rumusan masalah sebagai berikut:

1. Bagaimana cara untuk merancang purwarupa sistem otomasi penyiraman atau pemberian pupuk cair kepada tanaman dari jarak jauh dengan IoT?
2. Bagaimana cara untuk membangun purwarupa sistem otomasi penyiraman atau pemberian pupuk cair kepada tanaman dari jarak jauh dengan IoT?
3. Bagaimana cara kerja purwarupa sistem otomasi penyiraman atau pemberian pupuk cair kepada tanaman dari jarak jauh dengan IoT?
4. Bagaimana tingkat keberhasilan purwarupa sistem otomasi yang dibangun?

1.4 Batasan Masalah

Adapun batas masalah yang terkait dengan penelitian ini adalah:

1. Perangkat yang digunakan adalah Arduino Uno sebagai *microcontroller*, Nema 17 *stepper motor* sebagai motor penggerak, dan LoRa modul SX1278 sebagai *receiver* data.
2. Data yang memengaruhi otomasi adalah nilai kelembaban tanah dan/atau kadar pH atanah.
3. Otomasi yang dilakukan berupa penyiraman atau pemberian pupuk cair kepada tanaman.
4. Bahasa pemrograman yang digunakan adalah C++ yang sudah *embedded* di Arduino IDE.
5. Pengujian alat dilakukan pada *greenbox* yang berisi 3 tanaman cabai.
6. Pengambilan data dilakukan berdasarkan pertumbuhan tanaman cabai rawit dari awal penanaman.
7. Tidak membahas kemanan jaringan.

1.5 Metode Penelitian

Penelitian ini dilaksanakan berdasarkan beberapa metode yang dilakukan secara sistematis, yaitu sebagai berikut:

1. Studi Literatur

Pengumpulan materi dan referensi berupa buku, jurnal, artikel, *paper*, *slide* materi perkuliahan, dan lain-lain yang berkaitan dengan penelitian tugas akhir ini.

2. Studi Lapangan

Melakukan diskusi dengan dosen pembimbing dan juga ahli dalam bidang IoT yang dapat memberikan masukan untuk penelitian tugas akhir ini.

3. Perancangan dan Realisasi Sistem

Merancang kerangka kerja sistem lalu merealisasikan alat otomasi untuk *monitoring* dan *controlling* tanaman dari jarak jauh dengan IoT.

4. Implementasi Sistem

Mengimplementasikan alat otomasi untuk *monitoring* dan *controlling* pertumbuhan tanaman cabai berdasarkan parameter yang sudah ditentukan.

5. Analisis Kinerja Sistem

Menganalisis kinerja alat otomasi untuk *monitoring* dan *controlling* tanaman dari jarak jauh dengan IoT.

6. Kesimpulan

Menyimpulkan hasil akhir dari penelitian yang sudah dilakukan berdasarkan pada pertumbuhan tanaman cabai menggunakan alat otomasi untuk *monitoring* dan *controlling* tanaman dari jarak jauh dengan IoT.

BAB II

TINJAUAN PUSTAKA

2.1 *Internet of Things*

Secara bahasa, *Internet of Things* yang biasa disingkat IoT memiliki arti internet untuk banyak hal. Konsep dari IoT ini sendiri adalah dengan menghubungkan benda-benda di sekitar kita dengan internet sehingga bersifat *realtime*[4]. Dengan menghubungkan benda-benda di sekitar kita dengan internet, kita dapat memantau suatu keadaan dan juga mengendalikan benda-benda tersebut dari jarak jauh.

Munculnya konsep IoT ini pastinya diawali dengan penemuan internet. Internet mulai populer pada tahun 1989. Di tahun 1990, sudah ada peneliti bernama John Romkey yang berhasil menciptakan sebuah alat berupa pemanggang roti yang dapat dinyalakan dan dimatikan dengan menggunakan internet. Konsep alat tersebut sangat menyerupai konsep IoT. Di tahun 1997, Paul Saffo mulai mempopulerkan teknologi sensor yang sekarang sangat banyak digunakan dalam penerapan konsep IoT. Hingga saat ini, IoT sangatlah cepat pertumbuhannya dan para pemilik perusahaan sudah banyak yang menaruh investasi besar terhadap proyek-proyek IoT.[5]

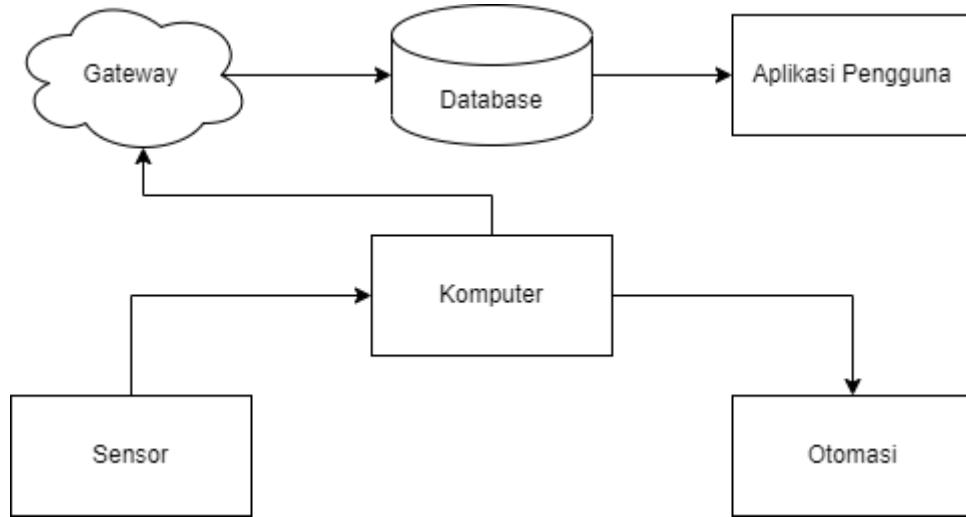
Benda-benda yang tadinya tidak terhubung dengan internet dan memiliki sistem komunikasinya tersendiri perlu ditanamkan sistem dari IoT ini yang biasa disebut dengan *embedded system*. Teknologi IoT ini membuat banyak hal dapat dipantau dan dikendalikan dengan mudah sehingga memberikan hasil yang optimal dengan usaha yang lebih sedikit. Teknologi ini sudah banyak diterapkan di berbagai bidang, contohnya adalah pertanian dan peternakan.

Terdapat tiga elemen penting yang ada di IoT, yaitu modul sensor, koneksi internet, dan pusat data yang disimpan dalam server[2]. Modul sensor berfungsi untuk merekam data dari objek yang dipantau, lalu data tersebut dikirimkan ke *database* yang ada di server melalui koneksi internet. Data yang terdapat di *database* dapat diakses secara *realtime* melalui *website* ataupun aplikasi *mobile*.

IoT juga memungkinkan untuk mengendalikan suatu perangkat dari jarak jauh yang dapat melakukan otomasi terhadap objek yang dipantau. Otomasi pada objek tersebut dapat dengan otomatis ataupun berdasarkan kehendak pengendali. Otomasi secara otomatis biasanya dilakukan jika ada suatu nilai dari parameter yang dipantau di bawah batas yang ditentukan sehingga dilakukan otomasi agar nilai tersebut dapat berubah sampai di interval yang diinginkan. Parameter tersebut ditentukan berdasarkan sensor yang dipasangkan terhadap objeknya.

Data yang telah direkam selama beberapa waktu dan dikumpulkan di *database* dapat diolah menjadi sebuah *dataset*. Di mana *dataset* tersebut dapat digunakan untuk menghasilkan model prediksi dengan metode *machine learning*. Metode tersebut dapat membuat perangkat menjadi pintar dengan sendirinya

sehingga perangkat tersebut di kemudian hari dapat mengambil aksi yang terbaik dengan sendirinya tanpa dikendalikan.



Gambar 1: Konsep Dasar Umum IoT

Konsep yang diterapkan di IoT dapat berbeda-beda tergantung kebutuhan masing-masing kondisi. *Gambar 1* di atas menjelaskan bahwa konsep dasar umum IoT yang dibawakan di penelitian ini terdapat komputer yang mendapatkan data dari sensor yang nantinya akan dikirimkan ke sistem otomasi dan juga database melalui gateway.

2.2 Tanaman Cabai Rawit

Cabai rawit adalah suatu tanaman berbuah dengan nama ilmiah *Capsicum Frutescens*. Cabai rawit adalah satu dari berbagai macam jenis cabai atau yang termasuk dalam genus *Capsicum*. Cabai merupakan salah satu komoditas tanaman yang paling laku di Indonesia. Cabai rawit memiliki ciri-ciri buah berwarna hijau kecil sewaktu muda dan berwarna merah tua jika telah masak.[3] Wujud dari cabai rawit dapat dilihat pada *Gambar 2* di bawah ini.



Gambar 2: Cabai Rawit

Tanaman cabai rawit memiliki tinggi sekitar 50 – 135 cm. Tanaman ini tumbuh tegak lurus ke atas. Akar cabai rawit merupakan akar tunggang. Akar tanaman ini umumnya melebar sejauh 30 – 50 cm secara vertikal dan dapat menembus tanah hingga kedalaman 60 cm. Batangnya kaku dan tidak bertrikoma. Daunnya merupakan daun tunggal yang bertangkai. Wujud dari tanaman cabai rawit dapat dilihat pada *Gambar 3* di bawah ini.



Gambar 3: Tanaman Cabai Rawit

Cabai rawit memiliki banyak kandungan gizi yang baik meskipun rasanya pedas. Kandungan tersebut meliputi kapsaisin, kapsantin, karotenid, alkaloid, resin, dan minyak atsiri. Cabai juga kaya akan kandungan vitamin A, B, dan C. Cabai

rawit merupakan jenis cabai yang paling banyak mengandung vitamin A dibanding dengan jenis cabai lainnya.

Tanaman cabai rawit tumbuh baik di tanah bertekstur lempung, lempung berpasir, dan lempung berdebu. Cabai rawit masih bisa tumbuh baik pada tekstur tanah yang agak berat, seperti lempung berlat. Tanah yang tidak baik untuk ditanami cabai rawit adalah tanah yang strukturnya padat dan tidak berongga. Tanah semacam ini akan sulit ditembus air pada saat penyiraman sehingga air akan tergenang. Jenis tanah dengan struktur padat dan tidak berongga di antaranya adalah tanah liat, tanah berkaolin, tanah berbatu, dan tanah berpasir.

Tanaman cabai rawit memiliki sifat adaptasi paling tinggi di antara jenis-jenis cabai lainnya. Hal ini membuat tanaman cabai dapat ditanam di dataran rendah hingga dataran tinggi. Namun, cabai rawit yang ditanam di dataran tinggi akan mengalami umur dan masa panen yang lebih lama dibanding dataran rendah, tetapi kualitas hasil panennya masih relatif sama.

Salah satu parameter yang memengaruhi pertumbuhan tanaman cabai rawit ataupun tanaman lain pada umumnya adalah tingkat kelembaban tanah (*soil moisture*). Kelembaban tanah adalah air yang mengisi sebagian atau seluruh pori-pori tanah yang berada di suatu tempat[6]. Curah hujan, jenis tanah, dan laju evapotranspirasi merupakan faktor-faktor yang menentukan kelembaban tanah yang akan menentukan ketersediaan air dalam tanah bagi pertumbuhan tanaman. Kelembaban tanah biasa diekspresikan dengan persentase dari *relative humidity* (RH). RH adalah perbandingan kadar air terhadap volume tanah, atau dapat diekspresikan dengan inchi air per kaki tanah.

Tingkat keasaman tanah yang biasa disebut pH tanah juga memengaruhi pertumbuhan dari tanaman cabai rawit ini. Tanaman cabai rawit dapat tumbuh optimal jika pH tanahnya di rentang 5,5 – 6,5. Jika pH tanah kurang dari 5,5, maka tanah harus diberi kapur pertanian[3]. Karena pada tanah yang asam, tanaman akan sulit untuk menyerap unsur hara, serta terdapat beberapa jamur yang dapat menimbulkan penyakit untuk tanaman.

Tanaman cabai rawit memerlukan lokasi lahan yang terbuka agar memperoleh peninjauan cahaya dari matahari dari pagi hingga sore hari. Selain itu, tanaman ini menyukai sistem drainase yang lancar, terutama pada saat musim hujan. Tanaman yang kurang cahaya akan mempunyai jumlah sel lebih sedikit dan habitus lebih tinggi daripada tanaman yang memperoleh banyak cahaya.

2.3 *Greenhouse In a Box / Greenbox*

Greenhouse memiliki arti ‘rumah hijau’ dalam Bahasa Indonesia. Sesuai namanya, *greenhouse* berfungsi sebagai wadah untuk tanaman yang mayoritas berwarna hijau agar dapat tumbuh dengan optimal. *Greenhouse* biasanya dibangun dengan mayoritas bahan kaca.

Ukuran dari luas *greenhouse* adalah beragam, mulai dari yang kecil sekitar 3 x 3 m hingga yang sangat luas sekitar ratusan meter persegi. *Greenhouse* dapat

menghindarkan tanaman yang ada di dalamnya dari debu, hujan, dan gangguan lainnya karena bersifat tertutup. Meskipun demikian, cahaya matahari tetap bisa masuk karena bahannya adalah kaca dan udara juga tetap bisa masuk dengan sirkulasi udara yang sudah diatur.



Gambar 4: Contoh Greenhouse Pada Umumnya

Gambar 4 di atas merupakan contoh *greenhouse* pada umumnya. Keberadaan *greenhouse* ini memungkinkan untuk memanipulasi kondisi lingkungan di dalam sehingga didapatkan kondisi yang ideal[7]. Manipulasinya dapat berupa menambahkan peralatan yang dapat mengatur kelembaban tanah, kelembaban suhu, intensitas cahaya, intensitas pengairan, dan lain-lain. Hal ini sangat berhubungan dengan penelitian ini di mana kita dapat menerapkan sistem IoT di perangkat-perangkatnya sehingga tercapai hasil yang optimal.

Dalam penelitian ini, ukuran wadah untuk penempatan tanaman-tanaman adalah 90 cm panjang, 30 cm lebar, dan 40 cm tinggi. Karena ukuran yang sangat kecil sehingga dapat dikategorikan sebagai *mini greenbox*, kami menamainya sebagai *greenbox*. *Greenbox* yang dibangun terbuat dari bahan akrilik dengan ketebalan 2 mm. *Greenbox* dalam penelitian ini sangat cocok untuk ditempatkan di dalam rumah karena ukurannya yang kecil.

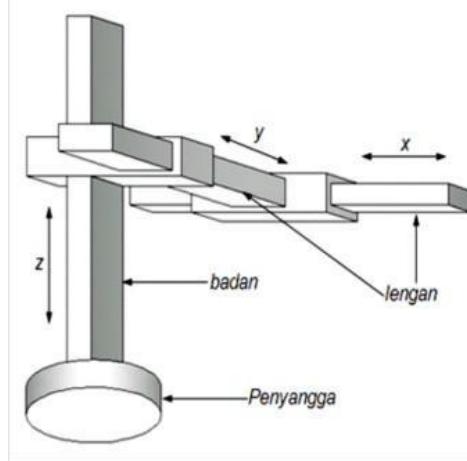
2.4 Robot

Robot adalah seperangkat alat mekanik yang bisa melakukan tugas fisik, baik dengan pengawasan dan kontrol manusia, ataupun menggunakan program yang telah didefinisikan terlebih dahulu (kecerdasan buatan)[8]. Istilah robot berawal bahasa Ceko “*robota*” yang berarti pekerja atau kuli yang tidak mengenal lelah atau bosan. Robot biasanya digunakan untuk tugas yang berat, berbahaya, pekerjaan yang berulang dan kotor. Biasanya kebanyakan robot industri digunakan dalam bidang produksi.

2.4.1 Robot Koordinat Kartesius

Robot Kartesius atau robot koordinat kartesius yang juga dikenal sebagai robot linier adalah robot industri dengan tiga sumbu kontrol utama yang semuanya linier (bergerak sepanjang garis lurus daripada berputar) dan saling tegak lurus satu sama lain. Robot kartesian adalah metode yang disukai untuk membuat gerakan titik-ke-titik, tetapi juga dapat melakukan gerakan

interpolasi dan berkontur yang kompleks. Jenis gerakan yang diperlukan akan menentukan perangkat kontrol terbaik, protokol jaringan, HMI, dan komponen gerakan lainnya untuk sistem[9].



Gambar 5: Konsep Dasar Robot Kartesius 3 Sumbu

Gambar 5 di atas menunjukkan bahwa robot kartesius dapat memiliki sumbu hingga 3 buah yang menjadikannya sebuah mesin yang bergerak tiga dimensi. Sumbu tersebut meliputi X, Y, dan Z. Namun, penelitian ini akan hanya menggunakan dua sumbu, yakni X dan Y karena otomasi penyiraman tidak memerlukan pergerakan sumbu Z. Terdapat badan, lengan, dan penyangga sebagai komponen utama dari robot kartesius secara umum. Gambar 6 dan Gambar 7 di bawah menunjukkan bahwa terdapat dua desain yang umum diterapkan pada robot kartesius.



Gambar 6: Desain Umum 1 Robot Kartesius



Gambar 7: Desain Umum 2 Robot Kartesius

2.4.2 *Computer Numerical Control (CNC)*

Istilah CNC adalah singkatan dari “Computer Numerical Control” yang dalam Bahasa Indonesia berarti komputer kontrol numerik. CNC merupakan sistem otomasi mesin perkakas yang dioperasikan oleh perintah yang diprogram secara abstrak dan disimpan di media penyimpanan. Kelebihan yang paling dominan yaitu kecepatan dalam proses produksi sehingga cocok digunakan untuk produksi massal.

Mesin CNC pertama diciptakan pertama kali pada tahun 1940-an hingga 1950-an dengan memodifikasi mesin perkakas biasa. Pada awalnya, mesin ini diperuntukkan untuk membuat benda kerja yang rumit. Namun, karena biaya pembuatan dan volume unit pengendali yang besar, hanya sedikit perusahaan yang mau berinvestasi dalam pengembangan teknologi ini. Tahun 1975 adalah tahun di mana mesin kontrol otomatis ini mulai berkembang pesat karena sudah adanya mikroprosesor sehingga volume unit pengendali dapat diperkecil dan disederhanakan. Saat ini, mesin CNC sudah banyak dipergunakan di segala bidang, seperti di bidang pendidikan, riset, serta tentunya industri-industri berskala nasional/internasional.[10]

2.5 *Hardware*

Hardware yang dalam Bahasa Indonesia adalah perangkat keras komputer merupakan semua jenis piranti atau komponen komputer yang bagian fisiknya dapat dilihat secara kasat mata dan dirasakan langsung. Terdapat jenis-jenis *hardware* menurut fungsinya, yaitu sebagai masukan, proses, keluaran, dan penyimpanan data. Dalam pembangunan purwarupa penelitian ini, membutuhkan beberapa *hardware* yang di antaranya ada *microcontroller*, *stepper motor*, *motor driver*, *CNC shield*, LoRa, antena, relay, transistor, dan resistor.

2.5.1 *Microcontroller*

Pengendali mikro *microcontroller* adalah sebuah komputer kecil yang dikemas dalam bentuk chip IC (*Integrated Circuit*) dan dirancang untuk melakukan tugas atau operasi tertentu. Pada dasarnya, sebuah IC *microcontroller* terdiri dari satu atau lebih inti prosesor (CPU), memori (RAM dan ROM), serta perangkat INPUT dan OUTPUT yang dapat diprogram. *Microcontroller* ini dapat diaplikasikan dalam produk ataupun perangkat yang

dikendalikan secara otomatis, seperti sistem kontrol mesin mobil, perangkat medis, pengendali jarak jauh, dan lain-lain.

Arduino adalah pengendali mikro *single-board* yang bersifat *open-source*, diturunkan dari *wiring platform*, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. Arduino Uno menggunakan prosesor Atmel AVR dan memiliki bahasa pemrograman sendiri. Saat ini Arduino sangat populer di seluruh dunia. Banyak pemula yang belajar mengenal robotika dan elektronika lewat Arduino karena mudah dipelajari. Bahasa yang dipakai dalam Arduino bukan *assembler* yang relatif sulit, tetapi bahasa C yang disederhanakan dengan bantuan pustaka-pustaka (*libraries*) Arduino.

Pada penelitian ini, penulis menggunakan *board* Arduino seri Arduino Uno R3. Arduino Uno adalah salah satu *development kit* pengendali mikro yang berbasis pada ATmega328P. Arduino Uno pertama kali dirilis pada tahun 2010. Terdapat *shield* Arduino Uno yang bermacam-macam membuat seri ini digemari banyak orang. Berikut ini merupakan spesifikasi dari *microcontroller* Arduino Uno yang tertera pada *Tabel 1* di bawah ini.

Tabel 1: Spesifikasi Arduino Uno (Arduino.cc 2022)

No	Nama	Nilai
1	Microcontroller	Atmega328P
2	Operating Voltage	5 V
3	Input Voltage	6 – 20 V
4	Digital I/O Pins	14
5	PWM Digital I/O Pins	6
6	Analog Input Pins	6
7	Flash Memory	32 KB
8	SRAM	2 KB
9	EEPROM	1 KB
10	Clock Speed	16 MHz
11	LED_BUILTIN	13
12	Weight	25 g
13	Dimension	68,6 x 53,4 mm

2.5.2 *Stepper Motor*

Stepper motor adalah motor DC yang bergerak dalam langkah diskrit. *Stepper motor* memiliki banyak kumparan yang diatur dalam kelompok yang disebut “fase”. Dengan memberi energi pada setiap fase secara berurutan,

motor akan berputar selangkah demi selangkah. Dengan langkah yang dikendalikan komputer, kita dapat mencapai posisi yang tepat dan/atau kontrol kecepatan. Untuk alasan ini, *stepper motor* adalah motor yang banyak dipilih untuk aplikasi kontrol gerakan presisi. *Stepper motor* mengubah pulsa-pulsa listrik yang diberikan menjadi gerakan-gerakan diskrit motor yang disebut langkah (step). Nilai dari suatu *stepper motor* diberikan dalam langkah per putaran (steps per revolution). *Stepper motor* umumnya mempunyai kecepatan dan torsi yang rendah.

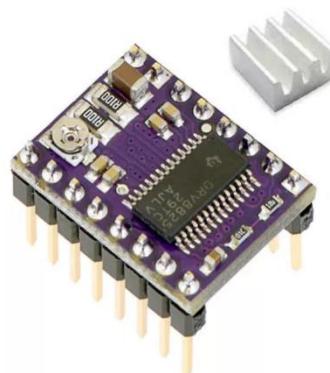
Stepper motor bekerja berdasarkan pulsa-pulsa yang diberikan pada lilitan fasenya dalam urutan yang tepat. Selain itu, pulsa-pulsa itu harus juga menyediakan arus yang cukup besar pada lilitan fase tersebut. Oleh karena itu, pengoperasian *stepper motor* harus mendesain terlebih dahulu suatu *sequencer logic* untuk menentukan urutan pencatuan lilitan fase motor yang kemudian menggunakan suatu penggerak (*driver*) untuk menyediakan arus yang dibutuhkan oleh lilitan fase. *Stepper motor* yang digunakan dalam pembangunan purwarupa penelitian ini adalah jenis Nema 17 yang menggunakan *motor driver* jenis DRV8825 dengan spesifikasi *stepper motor* yang tertera pada *Tabel 2* di bawah ini.

Tabel 2: Spesifikasi Stepper Motor Nema 17 (Depoinovasi.com 2011)

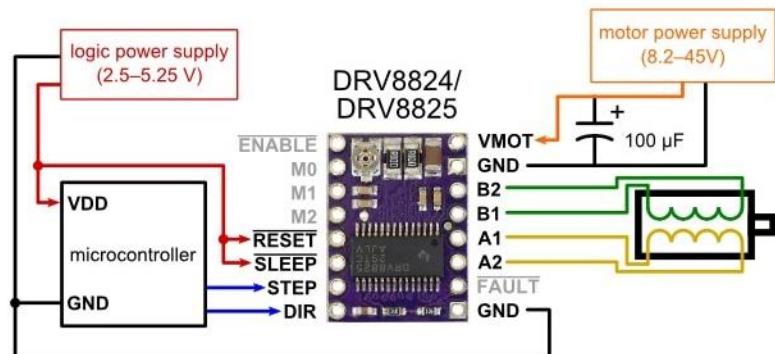
No	Nama	Nilai
1	Supply Voltage	12 V
2	Pulse per Rotation	200 P/R atau 1,8 deg/pulse
3	Phase Hybrid	2
4	Wire	4 Bipolar Wire
5	Holding Torque	7,4 Kg.cm
6	Shaft Diameter	5 mm
7	Timing Pulley	GT2-20 teeth
8	Bore Diameter	5 mm
9	Weight	550 g
10	Dimension	6,8 x 4,2 x 4,2 cm



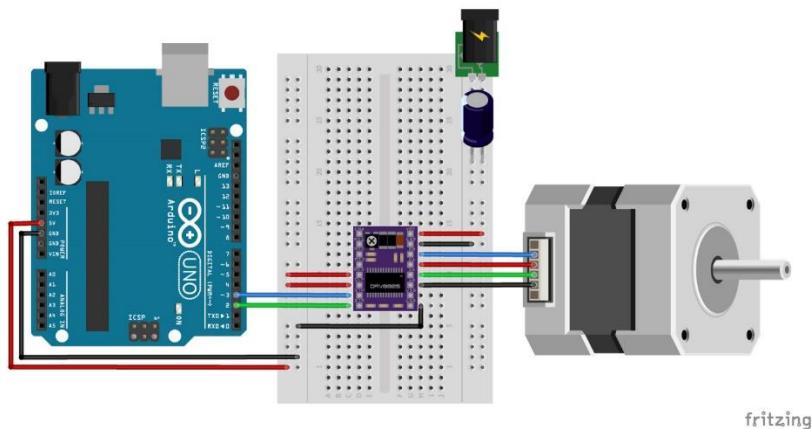
Gambar 8: Stepper Motor Nema 17



Gambar 9: Stepper Motor Driver DRV8825



Gambar 10: Denah Umum DRV8825

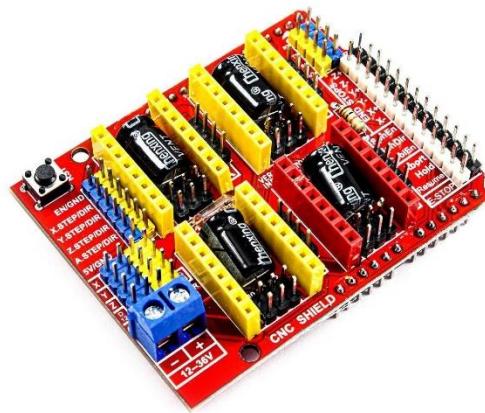


Gambar 11: Desain Umum Sistem Arduino Uno + DRV8825 + Stepper Motor

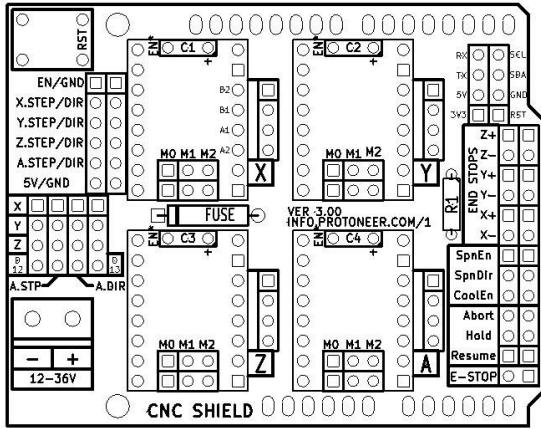
Gambar 8 di atas merupakan wujud dari *stepper motor* jenis Nema 17. Gambar 9 di atas merupakan wujud dari *stepper motor driver* jenis DRV8825. Gambar 10 di atas menjelaskan denah umum konseksi antara DRV8825 dengan *stepper motor*, *power supply*, dan *microcontroller*. Gambar 11 di atas menggambarkan desain umum rangkaian Arduino Uno yang dihubungkan dengan DRV8825 dan Nema 17 menggunakan *breadboard*.

2.5.3 Arduino CNC Shield

Arduino CNC *Shield* merupakan salah satu *shield* yang dibuat untuk dihubungkan dengan *board* Arduino. *Shield* adalah papan yang dapat dipasang di atas PCB Arduino untuk memperluas kemampuannya. *Shield* yang berbeda mengikuti filosofi yang sama dengan *toolkit* asli, yaitu mudah dipasang dan murah untuk diproduksi. *Shield* yang digunakan dalam rangkaian purwarupa penelitian ini adalah Arduino CNC *Shield* V3.0 karena akan mempermudah menghubungkan *board* Arduino dengan *stepper motor* dan *motor driver*.



Gambar 12: Arduino Uno CNC Shield V3



Gambar 13: Denah CNC Shield V3

Gambar 12 di atas merupakan wujud dari Arduino Uno CNC Shield V3. Sedangkan *Gambar 13* di atas adalah denah pin yang terdapat pada CNC Shield V3. Terdapat 4 slot untuk menempatkan *motor driver* serta beberapa pin lain untuk menyederhanakan rangkaian yang digunakan.

2.5.4 LoRa (*Long Range*)

LoRa (*Long Range*) adalah suatu format modulasi yang unik dan mengagumkan yang dibuat oleh Semtech[11]. Modulasi yang digunakan adalah modulasi FM. Inti pada pemrosesan menghasilkan nilai frekuensi yang stabil. Pada penelitian ini, jenis modul LoRa yang digunakan adalah SX1278. LoRa perlu terhubung dengan antena agar dapat memancarkan dan menerima sinyal atau yang biasa kita sebut sebagai *transceiver*. Penulis menggunakan antena *omnidirectional*. Antena *omnidirectional* memiliki pola radiasi ke segala arah dalam jumlah 360 derajat dengan daya yang sama, tetapi jangkauannya tidak terlalu jauh. Nilai frekuensi pada LoRa bermacam-macam sesuai daerahnya. Frekuensi 433 MHz biasa digunakan di Asia, 868 MHz di Eropa, 915 MHz di Amerika Utara[12]. Fitur-fitur yang tersedia di LoRa adalah:

1. **Geolocation**, fungsi ini memungkinkan kita dapat mendeteksi lokasi keberadaan suatu benda tanpa biaya.
2. **Biaya Rendah**, dapat mengurangi biaya dengan cara mengurangi biaya infrastruktur, biaya operasional, dan sensor-sensor yang mempunyai jaringanya sendiri.
3. **Terstandar**, dibuat agar dapat berinteraksi dan berfungsi dengan produk atau sistem lain sehingga dapat cepat beradaptasi dengan jaringan dan aplikasi IoT.
4. **Daya Rendah**, dengan konsumsi daya yang dibutuhkan hanya berkisar dari 13mA hingga 15mA sehingga baterai dapat bertahan dari 10 hingga 20 tahun.

5. **Jarak Jauh**, satu unit LoRa dapat memancarkan hingga 100 Km.
6. **Aman**, tertanam *end-to-end* enkripsi AES128.
7. **Kapasitas Tinggi**, Mendukung jutaan pesan per *base station*, ideal untuk operator jaringan publik yang melayani banyak pelanggan.

2.5.5 Relay, Transistor, Resistor

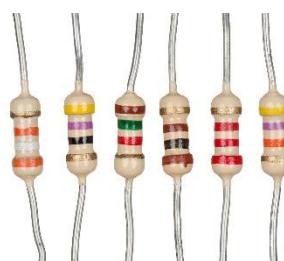
Relay, transistor, dan resistor adalah suatu komponen rangkaian elektronika yang sudah biasa digunakan. Relay berfungsi sebagai pengendali aliran listrik yang dapat dikontrol dengan program. Transistor memiliki berbagai macam fungsi di dalam rangkaian elektronika, ada pemutus, penyambung, modulasi tegangan, modulasi sinyal, serta dapat berfungsi sebagai kran listrik. Resistor atau hambatan merupakan komponen elektronika pasif yang memiliki fungsi guna menghambat serta mengatur arus listrik di dalam suatu rangkaian elektronika. Wujud dari relay, transistor, dan resistor dapat dilihat secara berturut-turut pada *Gambar 14*, *Gambar 15*, dan *Gambar 16* di bawah ini.



Gambar 14: Relay



Gambar 15: Transistor



Gambar 16: Resistor

2.6 Software

Software yang dalam Bahasa Indonesia adalah perangkat lunak komputer merupakan program digital yang dapat dijalankan untuk membuat atau mengerjakan sesuatu secara digital. *Software* biasanya dapat dijalankan jika dalam komputer apapun jenisnya. *Software* sendiri memiliki berbagai macam jenis dan fungsi, seperti mendesain, hiburan, simulasi, dan masih banyak lagi kegiatan lainnya. *Software* yang digunakan penulis pada penelitian kali ini ada dua, yaitu Arduino IDE dan Fritzing.

2.6.1 Arduino IDE

Arduino IDE (*Integrated Development Environment*) adalah *software* yang digunakan untuk memrogram *board* Arduino. Arduino IDE dapat diunduh secara gratis di *website* resmi Arduino IDE. Arduino IDE ini berguna sebagai *text editor* untuk membuat, mengubah, dan juga memvalidasi kode program. Arduino IDE juga digunakan untuk mengunggah program ke *board* Arduino. Kode program yang digunakan pada Arduino disebut dengan istilah Arduino *sketch* atau disebut juga *source code* dengan format ekstensi berupa “.ino”.

Bahasa pemrograman yang digunakan Arduino IDE adalah C++. Arduino IDE dipublikasikan sebagai *Open Source*, yang artinya tersedia bagi para *programmer* berpengalaman untuk pengembangan project lebih lanjut. Struktur wajib dalam Arduino IDE meliputi fungsi “*setup()*” yang merupakan fungsi yang akan dijalankan pada saat pertama kali dijalankan dan fungsi “*loop()*” yang merupakan fungsi yang akan dijalankan secara berulang-ulang. Arduino IDE juga dapat meng-*import library* yang bermacam-macam, kebanyakan *library* untuk perangkat-perangkat khusus *input* dan *output*. Hal ini membuat penggunaan Arduino cukup digemari oleh para pengembang alat.

2.6.2 Fritzing

Fritzing adalah suatu *software* atau perangkat lunak yang digunakan oleh desainer, seniman, dan para penghobi elektronika untuk perancangan berbagai peralatan elektronika. Fritzing memiliki daftar komponen yang cukup lengkap untuk dimasukkan ke dalam desain rangkaian. Fritzing juga dapat meng-*import* komponen *custom* yang belum ada di dalam daftar komponen bawaannya. Selain itu, Fritzing juga dapat menghadirkan desain rangkaian dalam mode *breadboard*, *schematic*, dan *PCB*, serta dapat melampirkan kode pemrograman.

BAB III

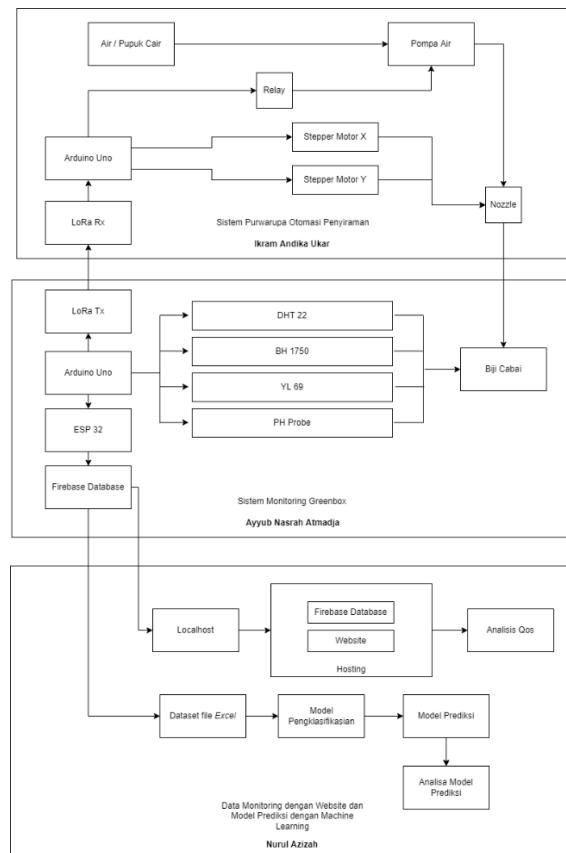
PERANCANGAN SISTEM

3.1 Konsep *Smart Greenbox*

Smart Greenbox memiliki konsep berupa sebuah box yang memiliki fungsi yang mirip seperti *greenhouse*, tetapi dengan ukuran yang jauh lebih kecil. Ukuran dari boxnya adalah 90x30x40 cm yang terbuat dari akrilik dengan ketebalan 2 mm. *Smart Greenbox* dirancang untuk dapat merawat tanaman dalam pot yang diletakkan di dalamnya. Terdapat sensor-sensor yang dipasangkan di tiap pot untuk mendeteksi intensitas cahaya, kelembaban udara, kelembaban tanah, dan pH tanah. Hasil dari sensor akan ditransmisikan ke *database* dan *microcontroller* alat otomasi. Jadi, alat otomasi akan melakukan penyiraman sesuai dengan masukan dari hasil sensor.

3.1.1 Diagram Blok Sistem *Smart Greenbox*

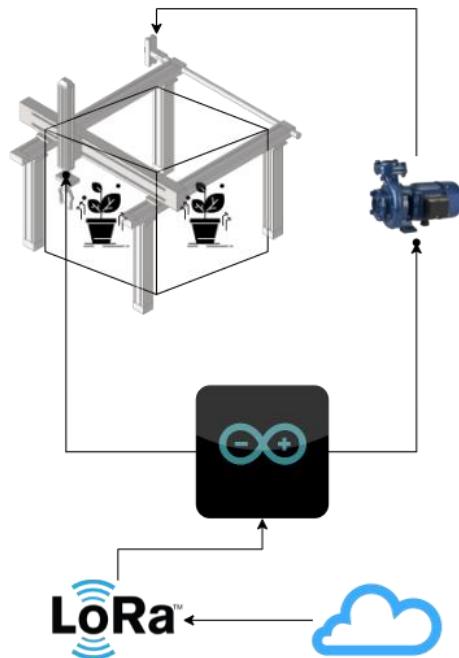
Pengerjaan sistem *Smart Greenbox* secara keseluruhan dibagi menjadi tiga. Penulis berperan sebagai pembangun purwarupa alat otomasi, Ayyub Nasrah Atmadja berperan sebagai pembangun perangkat sensor dan pengiriman data, dan Nurul Azizah berperan sebagai pembangun *website* dan pengolah data dengan *machine learning*.



Gambar 17: Diagram Blok Sistem *Smart Greenbox*

Berdasarkan *Gambar 17* di atas, penulis akan menerima data dari sistem milik Ayyub Nasrah Atmadja berupa nilai-nilai hasil sensor. Data yang didapat akan diurai untuk hanya mengambil nilai kelembaban tanah atau pH tanah saja yang nantinya dimasukkan ke dalam sebuah variabel. Lalu, dijalankan fungsi yang menentukan tanaman mana saja yang perlu disiram. Setelah itu, pengendali mikro akan memberikan perintah kepada motor penggerak alat dan pompa air agar melakukan penyiraman kepada tanaman tujuan.

3.1.2 Desain Sistem Purwarupa



Gambar 18: Desain Sistem Purwarupa Otomasi Penyiraman

Secara umum, desain sistem dari purwarupa ini adalah membuat alat robot kartesius sumbu X-Y dengan konsep CNC yang memiliki lengan berupa penyemprot agar dapat menyiram beberapa tanaman di posisi yang berbeda-beda seperti yang ditunjukkan pada *Gambar 18*. Purwarupa ini akan bergerak dan melakukan otomasi bergantung pada masukan yang diterima dari sensor YL-69. Sensor YL-69 akan mendeteksi nilai kelembaban tanah dengan rentang 0 – 1023 satuan. Nilai ideal dari kelembaban tanah adalah 25% – 50% dari RH (*relative humidity*). Maka dari itu, penulis menetapkan bahwa purwarupa akan menyemprotkan air ke tanaman jika nilai kelembaban tanah dari sensor YL-69 kurang dari 250 satuan.

3.1.3 Diagram Alir Kerja Sistem

Diagram alir adalah sebuah jenis diagram yang mewakili algoritma yang menampilkan langkah-langkah dalam bentuk simbol-simbol grafis dan urutannya dihubungkan dengan panah. Diagram ini mewakili ilustrasi atau penggambaran penyelesaian masalah. Diagram alir dapat membuat pembaca paham akan proses-proses yang dilalui sebuah sistem.



Gambar 19: Diagram Alir Kerja Otomasi Penyiraman

Gambar 19 di atas menjelaskan bahwa hal pertama yang dilakukan pada saat sistem ini dijalankan adalah mempersiapkan dan mengaktifkan alat. Setelah itu, *receiver* akan bersiap untuk menerima data dari *transmitter* yang jika data didapatkan, maka akan diurai untuk mendapatkan nilai yang diperlukan dan memasukkannya ke dalam variabel program. Terakhir, otomasi akan berjalan hanya jika minimal satu variabel yang nilainya di bawah standar yang ditentukan.

3.2 Pembangunan Purwarupa

Dalam membangun purwarupa, diperlukan rancangan desain, alat dan bahan, dan tahapan pembangunan alat. Desain dapat dilakukan menggunakan sketsa ataupun *software* perancangan perangkat. Alat dan bahan merupakan benda yang digunakan dalam tahapan pembangunan. Tahapan pembangunan merupakan pedoman untuk membangun alat secara sistematis agar pembangunan berjalan dengan lancar.

3.2.1 Alat dan Bahan Pembangunan Purwarupa

Alat adalah benda atau perangkat yang digunakan untuk melakukan pekerjaan, sedangkan bahan adalah benda yang dari mana sesuatu dapat dibuat darinya, atau barang yang dibutuhkan untuk membuat sesuatu. Berikut merupakan alat dan bahan dalam pembangunan purwarupa penelitian ini yang disusun dalam *Tabel 3* di bawah ini.

Tabel 3: Alat dan Bahan Pembangunan Purwarupa Otomasi Penyiraman

No	Alat dan Bahan	Jumlah	Fungsi
1	Arduino Uno	1 pcs	<i>Microcontroller</i>
2	CNC Shield	1 pcs	Mempermudah <i>wiring motor driver</i>
3	LoRa SX1278	1 pcs	<i>Transceiver</i> data
4	Stepper Motor Nema 17	2 pcs	Motor penggerak dengan rotasi yang dapat ditentukan
5	Motor Driver DRV8825	2 pcs	Penghubung dari <i>microcontroller</i> ke <i>stepper motor</i>
6	SC8UU	6 pcs	Dudukan di jalur besi X dan Y
7	GT2 Timing Belt	3 m	Jalur yang digerakkan oleh motor agar berputar
8	GT2 Pulley Timing 20T	2 pcs	Poros aktif
9	GT2 Idler Pulley 20T	2 pcs	Poros pasif
10	Bracket L Socket Nema 17	2 pcs	Dudukan <i>stepper motor</i>
11	SFH8 Shaft Holder	8 pcs	Penahan ujung jalur besi
12	Rod Steel	2,8 m	Jalur besi X dan Y
13	Besi Galvanum 4x2 cm	10 m	Kerangka purwarupa
14	Baut + Mur + Ring 5 mm	16 pcs	Penghubung antar komponen
15	Baut + Mur + Ring 3 mm	46 pcs	Penghubung antar komponen
16	Pompa DC Mollar 12 V	1 pcs	Pompa air

17	Nozzle Brass	1 pcs	Mulut semprotan
18	Selang air 5/16	2 – 3 m	Jalur air
19	Relay	1 pcs	On/Off tegangan
20	Transistor NPN	1 pcs	<i>Switch</i> agar memakai tegangan <i>supply</i>
21	Resistor 100 K	1 pcs	Untuk hambatan transistor
22	Resistor 1 K	1 pcs	Untuk hambatan transistor
23	Adapter 24 V	1 pcs	Adapter <i>stepper motor</i>
24	Adapter 12 V	1 pcs	Adapter pompa
25	Gerinda tangan	-	Memotong besi
26	Alat las	-	Membentuk besi
27	Bor listrik	-	Membuat bolongan di besi untuk dipasangkan baut
28	Mini Grinder	-	Menghaluskan potongan besi
29	Waterpass	-	Mengukur dengan presisi
30	Penggaris	-	Mengukur panjang
31	Penggaris siku	-	Mengukur sudut
32	Solder	-	Memanaskan timah
33	Timah	-	Dipanaskan untuk menghubungkan jalur elektronik
34	Multimeter	-	Menghitung tegangan, arus, dll dari rangkaian
35	Box X4 <i>Custom</i>	1	Wadah untuk menata rangkaian <i>board</i> agar rapi

3.2.2 Tahapan Pembangunan Purwarupa

Penulis membangun purwarupa dengan tahapan-tahapan sebagai berikut:

1. Perancangan desain dan ukuran
Proses ini dilakukan dengan cara mencari referensi desain dari berbagai sumber yang nantinya akan didesain ulang dengan detail sesuai yang kita inginkan.
2. Pengumpulan alat dan bahan
Proses ini dilakukan dengan cara membuat daftar keperluan alat dan bahan yang nantinya akan dikumpulkan untuk pembangunan purwarupa.
3. Pembuatan kerangka mesin CNC sumbu X-Y

Proses ini dilakukan dengan cara membuat kerangka CNC sumbu X-Y yang terdiri atas *stepper motor*, *timing belt*, *pulley*, *idler pulley*, *rod steel*, SC8UU, SFH8, dan *bracket L* sesuai dengan ukuran yang didesain.

4. Pembuatan kerangka besi

Proses ini dilakukan dengan cara memotong-motong besi galvanum dan merangkainya hingga menjadi kerangka sesuai dengan desain yang kita inginkan.

5. Penyambungan kerangka

Proses ini dilakukan dengan cara menyambung kerangka CNC dengan kerangka besi yang dihubungkan dengan set baut, mur, dan *ring*.

6. Pemasangan alat penyiraman air

Proses ini dilakukan dengan cara memasang *nozzle* di kerangka CNC dan menghubungkannya ke pompa dengan menggunakan selang.

7. Perakitan *board* Arduino

Proses ini dilakukan dengan cara merangkai *board* Arduino yang akan dihubungkan ke *stepper motor*, LoRa, dan pompa air.

8. Pemrograman

Proses ini dilakukan dengan cara membuat program di Arduino IDE untuk menerima data dari LoRa, menggerakan *stepper motor*, dan menyalakan pompa air yang akan ditanamkan ke Arduino Uno.

9. *Finishing* dan pengujian

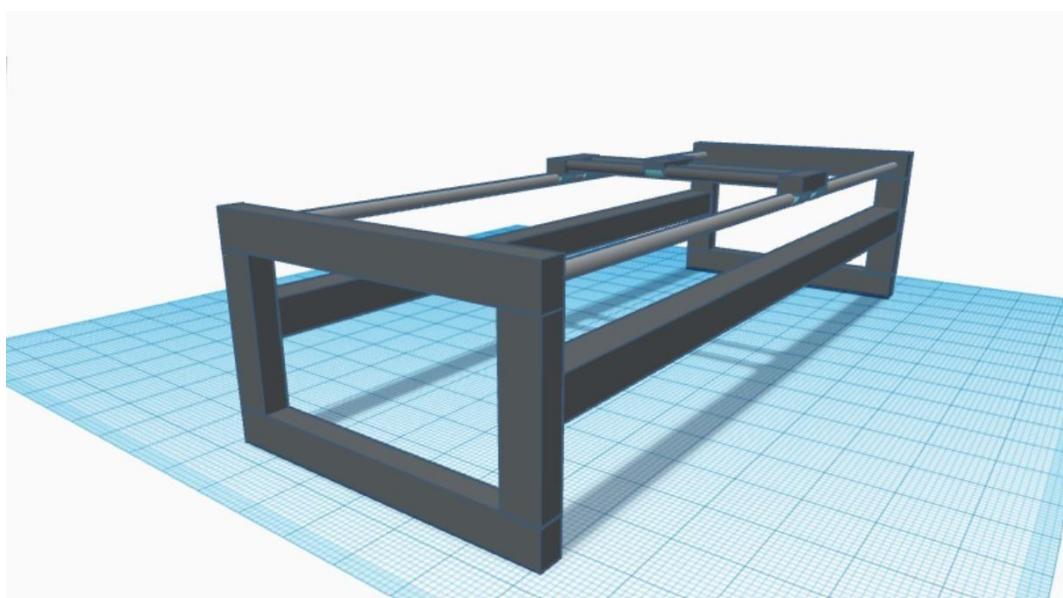
Proses ini dilakukan dengan cara menghubungkan semua komponen, merapikan kerangka dan rangkaian, dan melakukan pengujian sesuai metode yang dilakukan untuk dianalisis.

3.3 Desain Rangkaian Purwarupa

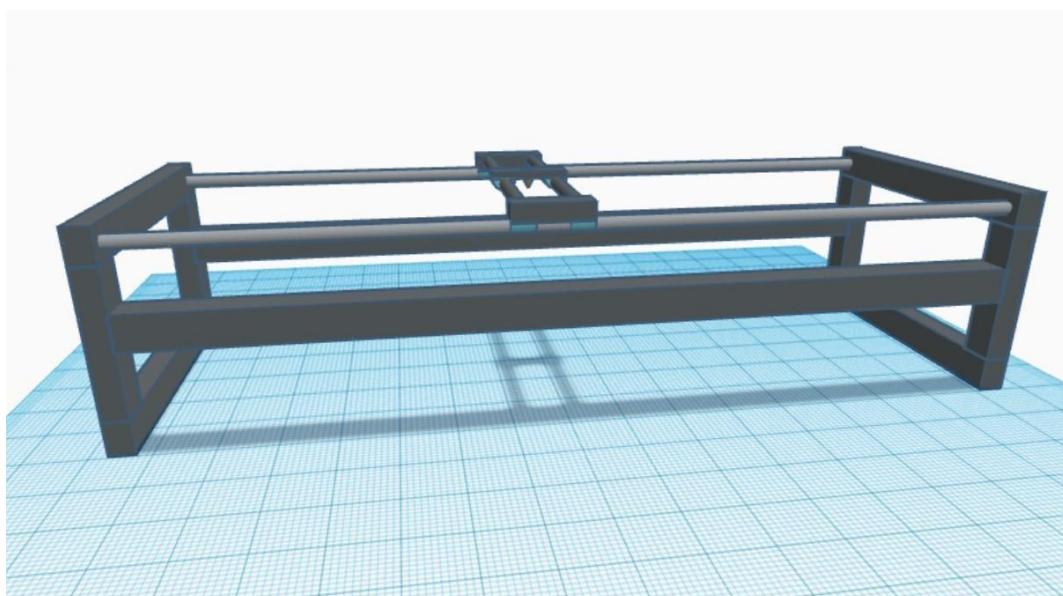
Desain rangkaian purwarupa diperlukan untuk mengetahui bentuk dari alat yang ingin dibangun serta gambaran rangkaian elektronika untuk mengaktifkan komponen-komponennya.

3.3.1 Desain Purwarupa

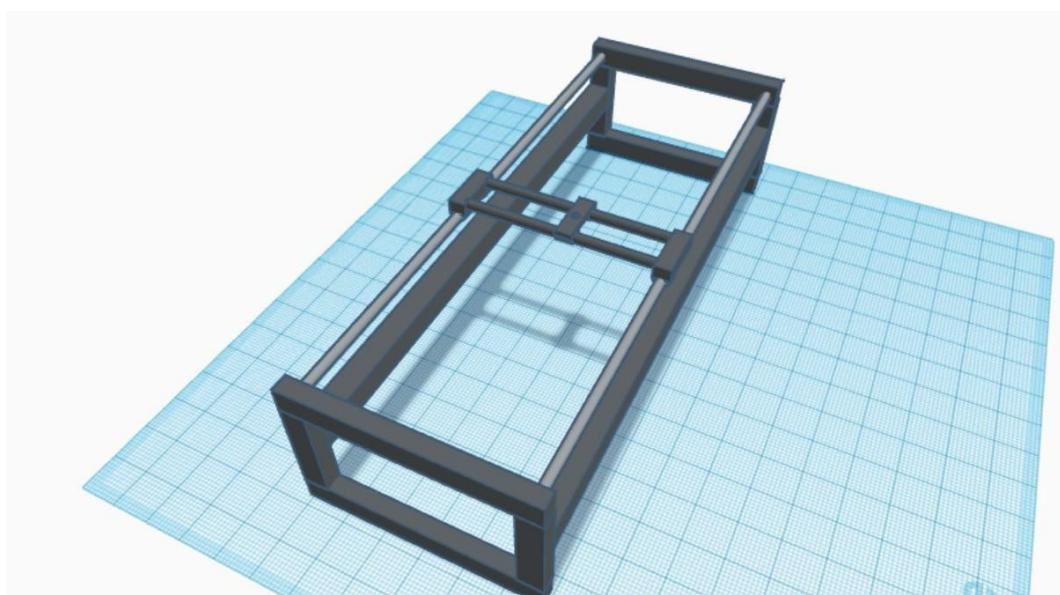
Dalam penelitian ini, penulis menggunakan website tinkercad.com untuk membuat desain dari kerangka alat berbasis 3D. Ukuran dimensi dari alat yang dibangun adalah 100 cm x 60 cm x 50 cm. Berikut merupakan hasil desain kerangka alat dari berbagai sudut pandang seperti pada *Gambar 20*, *Gambar 21*, *Gambar 22*, *Gambar 23*, dan *Gambar 24* di bawah ini.



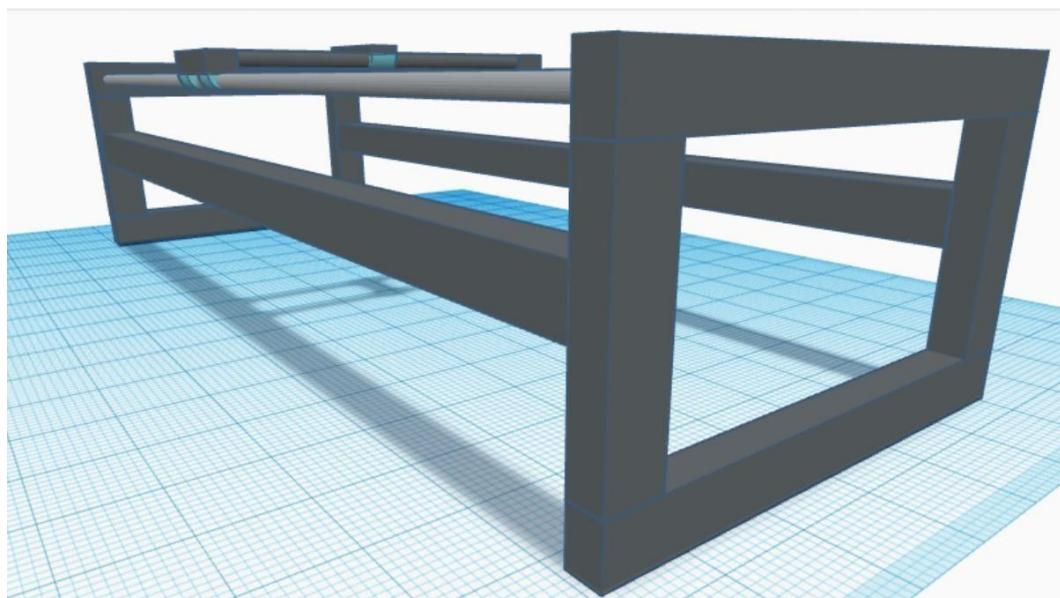
Gambar 20: Tampak 1 Desain Purwarupa



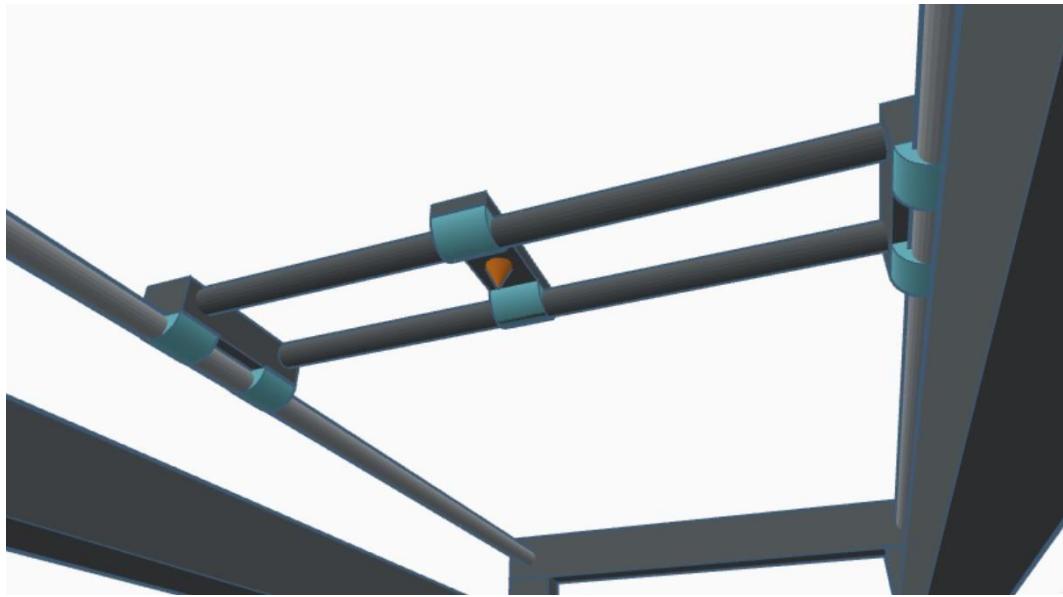
Gambar 21: Tampak 2 Desain Purwarupa



Gambar 22: Tampak 3 Desain Purwarupa



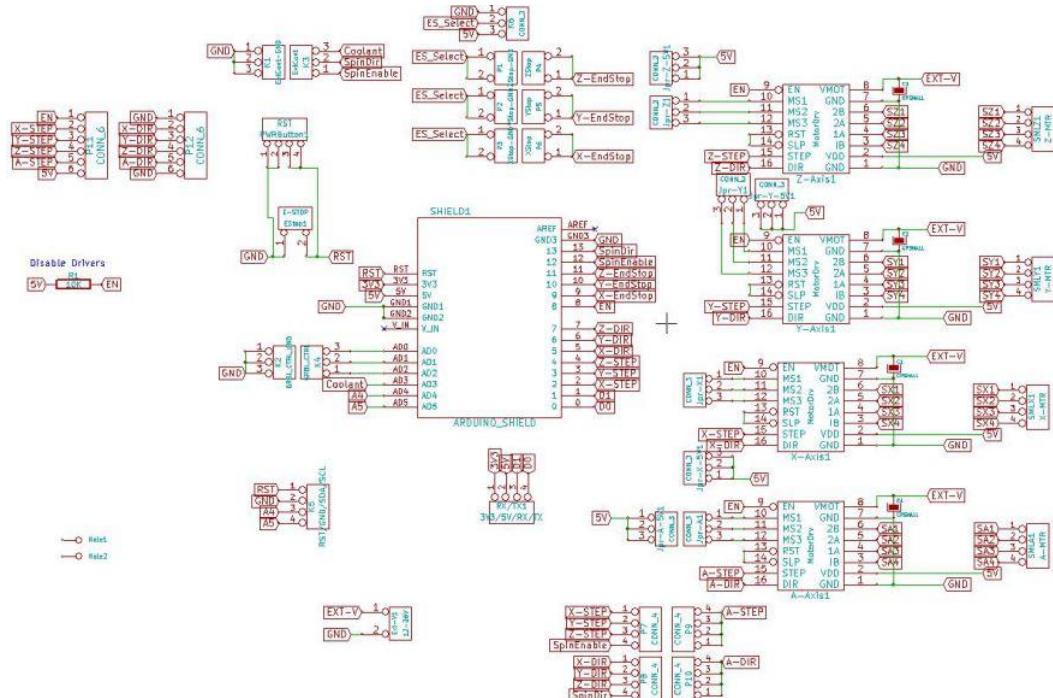
Gambar 23: Tampak 4 Desain Purwarupa



Gambar 24: Tampak 5 Desain Purwarupa

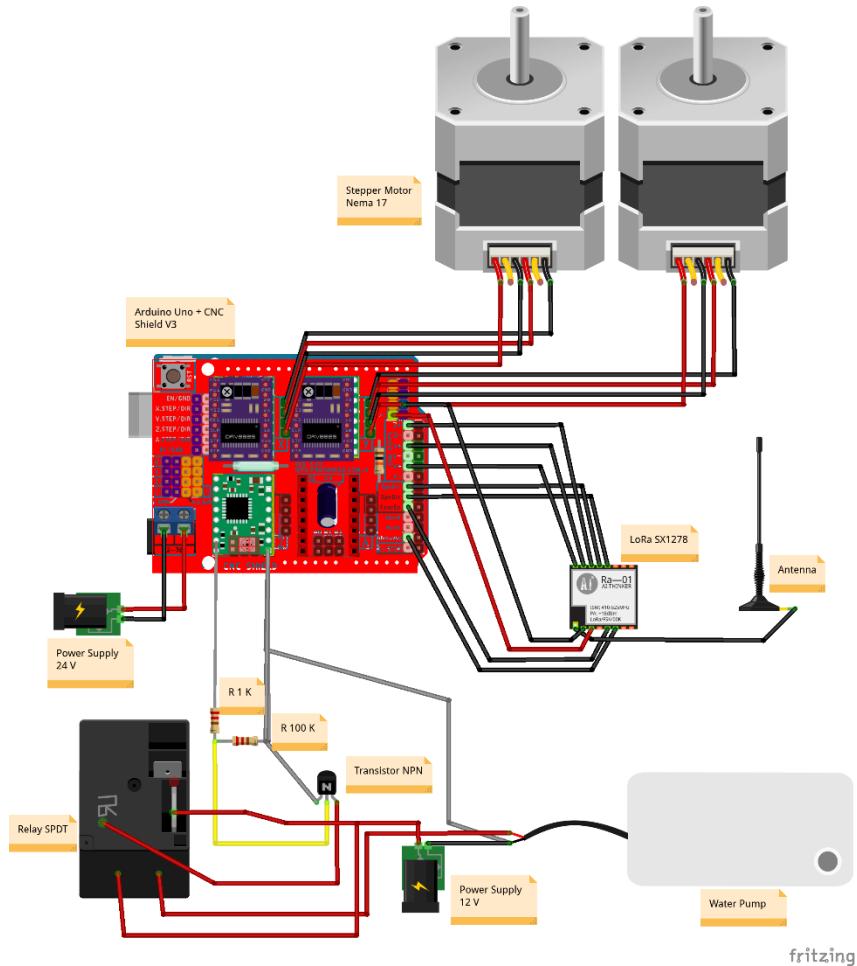
3.3.2 Rangkaian Purwarupa

Penulis merancang rangkaian komponen elektronika dari purwarupaini dengan menggunakan *software* Fritzing. Terdapat beberapa komponen yang dimasukkan ke dalam Fritzing, yakni Arduino Uno R3, CNC Shield V3, Nema 17, pompa air, LoRa, antena, kabel, *power supply*, resistor, transistor, dan relay.



Gambar 25: Rangkaian Skematik Arduino Uno - CNC Shield V3

Gambar 25 di atas menunjukkan pin-pin yang dapat terhubung di pemasangan CNC *shield* terhadap Arduino Uno. Alasan digunakannya CNC *shield* adalah untuk mempermudah pemasangan rangkaian karena CNC *shield* didesain khusus untuk pemasangan *motor driver* dan *stepper motor*. Jika tidak menggunakan CNC *shield*, maka rangkaian akan harus menggunakan *breadboard* yang di mana hal tersebut tidak efisien karena terdapat beberapa pin dari *motor driver* dan *stepper motor* yang perlu dihubungkan ke pin yang sama di Arduino Uno.



Gambar 26: Rangkaian Komponen Sistem Purwarupa

Tabel 4: Tata Hubung Pin Komponen LoRa dan Pompa Air

No	Pin Komponen	Arduino	CNC Shield
1	LoRa SCK	13	SpnDir
2	LoRa MISO	12	SpnEn
3	LoRa MOSI	11	Z+ Z-
4	LoRa NSS	10	Y+ Y-
5	LoRa RST	9	X+ X-
6	LoRa DIO0	17 (A3)	CoolEn
7	LoRa DIO1	16 (A2)	Resume
8	Relay	7	Z-Dir
9	Emitter Transistor	GND	GND

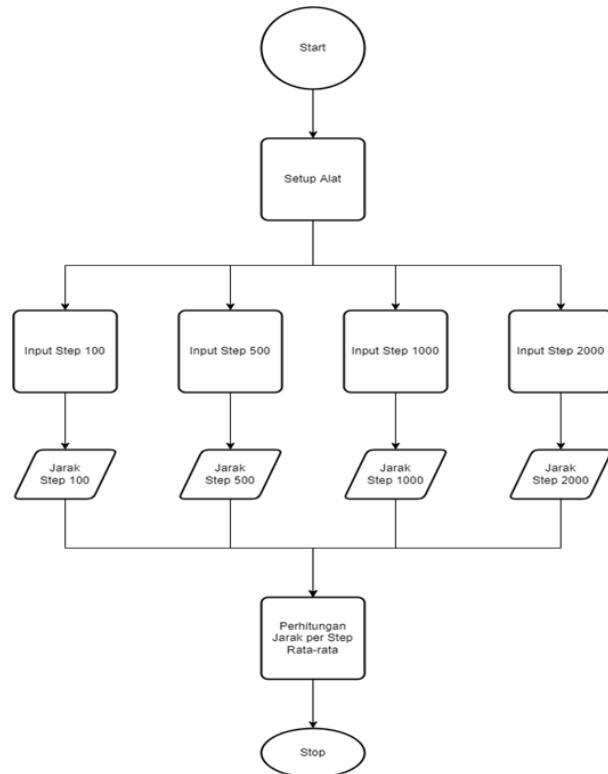
Gambar 26 merupakan desain rangkaian elektronika yang menjadi acuan untuk pembangunan purwarupa sistem otomasi pada penelitian ini. Karena CNC shield didesain khusus untuk penempatan *motor driver* dan penghubung dengan *stepper motor*, maka diperlukan tata hubung pin yang akan dihubungkan dengan LoRa dan pompa air seperti yang tertera pada *Tabel 4*. Jadi, pin pada LoRa dan pompa air yang diawali dengan relay tidak perlu dihubungkan langsung ke *board Arduino* yang sudah tertutup oleh CNC shield.

3.4 Metode Analisis Performansi Alat

Terdapat beberapa metode untuk menganalisis performansi alat. Pengujian yang pertama adalah pengujian posisi untuk menentukan nilai jarak berdasarkan *step*. Pengujian yang kedua adalah pengujian tingkat akurasi dan kemampuan untuk pengulangan alat. Pengujian yang terakhir adalah pengujian waktu otomasi berdasarkan dua pilihan algoritma pemrograman untuk menentukan efisiensi waktu dari program adan alat.

3.4.1 Pengujian Posisi

Pengujian posisi dilakukan untuk mengetahui jarak yang dihasilkan dari *step motor* berdasarkan kalimat pemrogramannya. Terdapat istilah *step* dalam konsep mekanika *stepper motor*. *Step* dapat diibaratkan sebagai ketukan yang diberikan oleh *stepper motor* terhadap porosnya untuk berputar. Berdasarkan hasil perbandingan beberapa input *step* yang menghasilkan output jarak oleh *stepper motor*, maka diperoleh jarak rata-rata per *step*. Tahapan untuk pengujian posisi ini divisualisasikan dengan diagram alir seperti *Gambar 27* di bawah ini.



Gambar 27: Diagram Alir Pengujian Posisi

3.4.2 Pengujian Akurasi & Repeatability

Pengujian akurasi dan *repeatability* dilakukan untuk mengetahui tingkat akurasi input pemrograman terhadap pergerakan otomasi, serta mengetahui rentang konsistensi alat dalam melakukan otomasi yang berulang. Pengujian ini dilakukan dengan cara menentukan terlebih dahulu posisi yang hendak diberlakukan otomasi, lalu melakukan otomasi kepada target posisi tersebut selama berulang kali. Hasil yang didapatkan berupa persentase akurasi dan nilai toleransi alat. Tahapan untuk pengujian akurasi & *repeatability* ini divisualisasikan dengan diagram alir seperti *Gambar 28* di bawah ini.



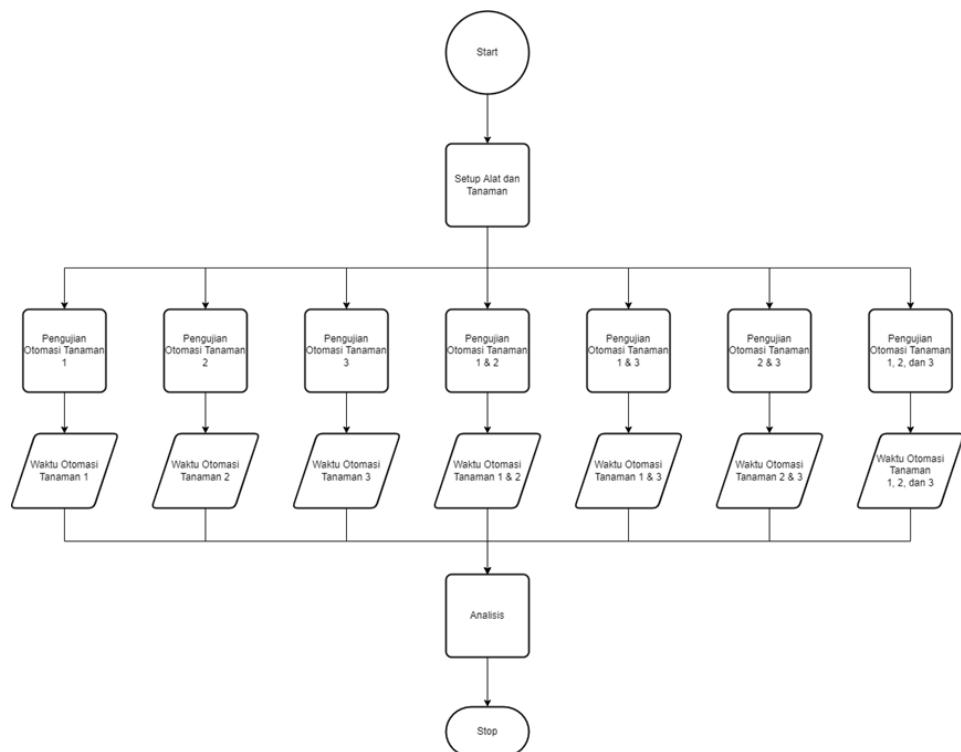
Gambar 28: Diagram Alir Pengujian Akurasi dan Repeatability

3.4.3 Pengujian Waktu Proses Otomasi

Pengujian ini dilakukan untuk mengetahui lamanya sistem untuk melakukan otomasi terhadap beberapa jumlah tanaman di posisi yang berbeda. Pengujian ini dilakukan dengan cara menghitung waktu otomasi terhadap:

1. Tanaman 1
2. Tanaman 2
3. Tanaman 3
4. Tanaman 1 dan 2
5. Tanaman 1 dan 3
6. Tanaman 2 dan 3
7. Tanaman 1, 2, dan 3

Tahapan untuk pengujian waktu proses otomasi ini divisualisasikan dengan diagram alir seperti *Gambar 29* di bawah ini.



Gambar 29: Diagram Alir Pengujian Waktu Proses Otomasi

3.5 Pengujian Parameter Pengiriman LoRa

Pengujian Parameter dilakukan untuk mengetahui kualitas pengiriman data di sistem di mana terdapat pengirim dan penerima. Parameter yang diukur adalah pengiriman dari LoRa *board* sensor milik Ayyub Nasrah Atmadja kepada *board* otomasi milik penulis. Parameter yang diukur berupa RSSI (*Received Signal Strength Indicator*) dan SNR (*Signal Noise Ratio*).

3.5.1 RSSI (*Received Signal Strength Indicator*)

RSSI merupakan indikator penanda kekuatan sinyal yang dapat terekap oleh *receiver*. Baik tidaknya suatu RSSI dinilai dari seberapa jauh RSSI yang terekap dari nilai 0. Semakin dekat nilai RSSI ke 0, maka RSSI dapat dikatakan baik. Sebaliknya, apabila semakin jauh nilai RSSI ke 0, maka RSSI dapat dikatakan buruk. Nilai RSSI yang baik berkisar antara -1 hingga -99 dBm, sedangkan nilai RSSI yang buruk berada di bawah -100 dBm[11].

$$\text{Rata-Rata RSSI} = \frac{\text{Jumlah RSSI}}{\text{Paket Diterima}} \quad (1)$$

Rata-Rata RSSI dapat ditentukan melalui pembagian antara Jumlah RSSI dengan Jumlah Paket Diterima seperti yang tertulis pada persamaan (1). RSSI dapat diperoleh dari program Arduino melalui *library* “LoraLib.h” yang secara otomatis dapat menampilkan nilai RSSI.

3.5.2 SNR (*Signal Noise Ratio*)

SNR merupakan perbandingan antara kekuatan sinyal dengan *noise level*. Semakin besar nilai SNR, maka semakin baik kualitas jalan dari pengiriman data. Artinya, semakin besar pula kemungkinan jalan tersebut dipakai untuk lalu lintas komunikasi informasi dan sinyal dalam kecepatan tinggi. Nilai SNR dapat dikatakan cenderung stabil, tidak bergantung pada berapapun kecepatan informasi yang melewati jalan tersebut. Efek yang ditimbulkan dari SNR yang buruk adalah berupa koneksi lambat dan sering terputus[11]. Berikut merupakan kategori kualitas SNR, yaitu:

- $\geq 29,0 \text{ dB}$ = *Outstanding* (bagus sekali),
- $20,0 \text{ dB} - 28,9 \text{ dB}$ = *Excellent* (bagus, koneksi stabil),
- $11,0 \text{ dB} - 19,9 \text{ dB}$ = *Good* (baik, sinkronisasi sinusal ADSL dapat berlangsung lancar),
- $7,0 \text{ dB} - 10,9 \text{ dB}$ = *Fair* (cukup, rentan terhadap variasi perubahan kondisi pada jaringan),
- $0,0 \text{ dB} - 6,9 \text{ dB}$ = *Bad* (buruk, sinkronisasi sinyal tidak lancar).

$$\text{Rata-Rata SNR} = \frac{\text{Jumlah SNR}}{\text{Paket Diterima}} \quad (2)$$

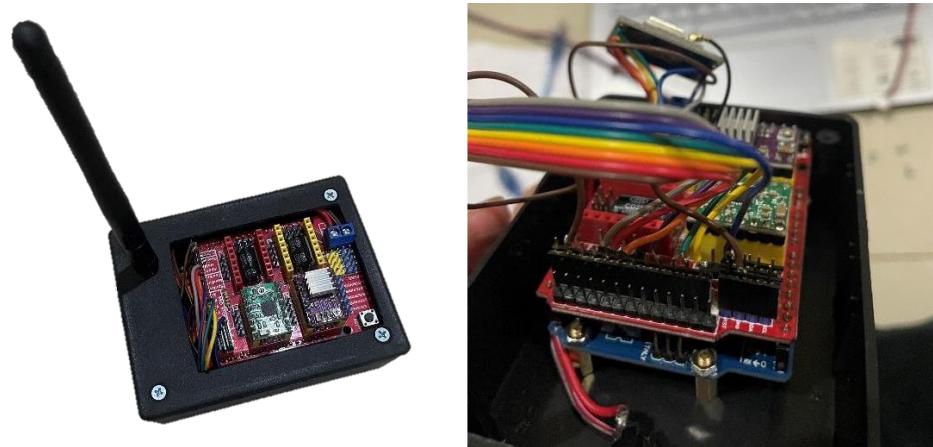
Rata-rata SNR dapat ditentukan melalui pembagian antara Jumlah SNR dengan Jumlah Paket Diterima seperti yang tertulis pada persamaan (2). SNR dapat diperoleh dari program Arduino melalui library “LoraLib.h” yang secara otomatis dapat menampilkan nilai RSSI.

BAB IV

HASIL KERJA SISTEM

4.1 Hasil Pembangunan Alat

Purwarupa alat telah berhasil dibangun dalam kurun waktu sekitar 3 bulan lebih terhitung dari bulan Oktober 2021 hingga bulan Januari 2022. Ukuran dimensi dari purwarupa adalah 104,5 cm x 59 cm x 50 cm diukur menggunakan mistar. Berikut adalah hasil dokumentasi dari purwarupa alat otomasi penyiraman terhadap *greenbox*.



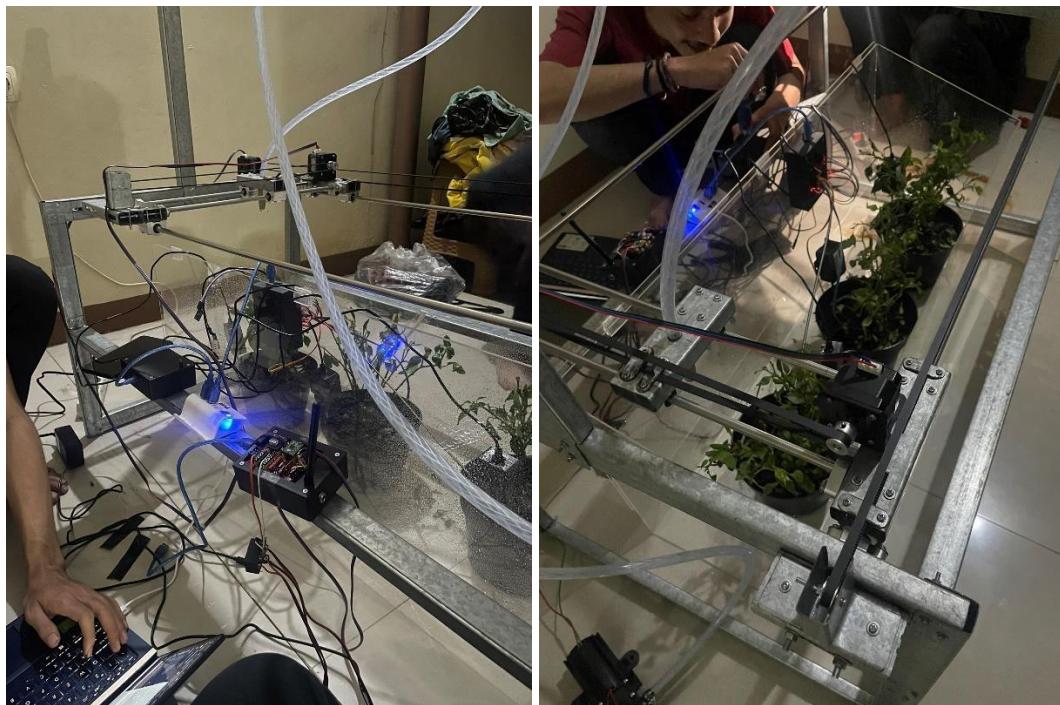
Gambar 30: Rangkaian Arduino Uno + CNC Shield + LoRa di dalam Box X4 Custom

Gambar 30 di atas menunjukkan bahwa rangkaian dimasukkan ke dalam Box X4 custom agar terlihat rapi. Rangkaian yang ada di dalam kotak tersebut meliputi Arduino Uno, CNC Shield, dan LoRa. Komponen lain seperti kabel penghubung ke rangkaian ke *stepper motor* dan pompa air diletakkan di luar kotak.



Gambar 31: Kerangka Besi dan Mesin CNC Purwarupa

Gambar 31 di atas menunjukkan kerangka besi yang sudah jadi dan mesin CNC yang sudah dipasangkan. Dapat dilihat juga bahwa ukuran dimensi dari purwarupa alat sedikit lebih besar dibandingkan dengan *greenbox* yang digunakan. Terdapat juga tambahan rangkaian besi yang terletak di atas untuk membantu mengikat selang dan kabel agar tidak ke mana-mana.



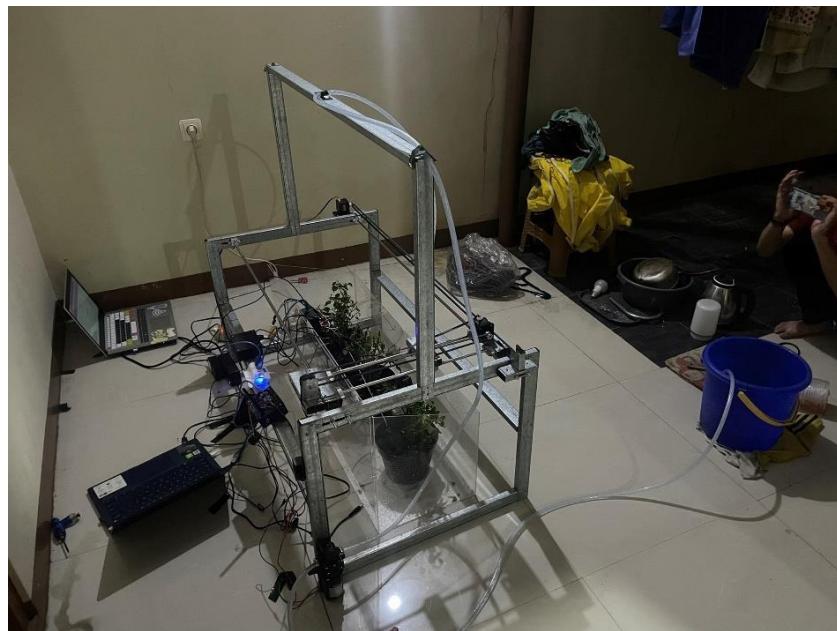
Gambar 32: Penempatan Board dan Tanaman

Gambar 32 di atas menunjukkan bahwa *board* sensor diletakkan di dalam *greenbox* sedangkan *board gateway* dan *board otomasi* diletakkan di kerangka besi yang berada di luar *greenbox*. Penempatan tanaman diletakkan sejauh 1000 *step* dari ujung jalur otomasi secara horizontal dan 1000 *step* antar tanaman. 1000 *step* setara dengan sekitar 20 cm berdasarkan hasil pengujian alat.



Gambar 33: Tampak Depan Sistem Smart Greenbox

Gambar 33 di atas menunjukkan seperti apa posisi tanaman ketika pengujian. *Nozzle* memiliki posisi awal di ujung kiri sumbu X dari tampak depan di mana berseberangan dengan *stepper motor* sumbu X. Sedangkan dari sumbu Y, *nozzle* memiliki posisi awal di tengah-tengah karena otomasi penyiraman terhadap *greenbox* yang berisi tiga pot tanaman cabai ini diletakkan secara linear sehingga tidak diperlukan pergerakan sumbu Y.



Gambar 34: Keseluruhan Sistem Smart Greenbox

Gambar 34 di atas menunjukkan keseluruhan sistem *Smart Greenbox* yang berisi sistem otomasi dan sistem sensor. Terlihat bahwa pompa akan menarik air dari ember berisi air dan mendorong air menuju *nozzle* melalui selang air.

4.2 Hasil Pengujian Posisi

Pengujian posisi telah dilakukan. Pengujian dilakukan dengan memberikan perintah untuk bergerak berdasarkan jumlah *step* tertentu dan lalu menghitung jarak yang dihasilkan. Pengujian dilakukan terhadap masing-masing sumbu karena jalur X dan Y terdapat perbedaan panjang. Berikut merupakan hasil dari pengujian posisi yang diukur dengan mistar.

Tabel 5: Hasil Pengujian Posisi Sumbu X

X (cm)		
Input (step)	Jarak (cm)	Jarak per Step (cm)
100	2,15	0,0215
500	10,45	0,0209
1000	20,45	0,02045
2000	40,3	0,02015
Jarak per Step Rata-rata		0,0210875

Tabel 6: Hasil Pengujian Posisi Sumbu Y

Y (cm)		
Input (step)	Jarak (cm)	Jarak per Step (cm)
100	2,1	0,021
500	10,3	0,0206
1000	20,2	0,0202
Jarak per Step Rata-rata		0,0206

Jarak per Step Rata-Rata didapatkan dari penjumlahan dari jarak masing-masing step lalu dibagi dengan banyaknya variasi input step. Dari hasil pengujian dan perhitungan yang tertera pada *Tabel 5* dan *Tabel 6*, didapatkan nilai Jarak per Step rata-rata sebesar 0,0210875 cm untuk sumbu X dan 0,0206 untuk sumbu Y. Jarak per Step rata-rata dapat berubah di tiap penelitian karena dipengaruhi oleh spesifikasi alat dan bahan yang digunakan, yaitu tegangan, arus, serta jenis dan ukuran *timing belt*, *pulley*, dan jalur besi yang digunakan.

4.3 Hasil Pengujian Akurasi dan *Repeatability*

Pengujian akurasi dan *repeatability* telah dilakukan dengan cara memberikan perintah kepada alat robot agar bergerak ke koordinat yang ditentukan dan diulang sebanyak 20 kali. Pengujian ini untuk mengetahui akurasi dari setiap pergerakan dari perintah dan juga kemampuan untuk mempertahankan konsistensi terhadap perintah yang berulang atau sama persis.

Telah dilakukan pengujian berupa melaksanakan otomasi terhadap posisi *nozzle* yang ditentukan dengan *step* sumbu X sebanyak 2000 dan *step* sumbu Y sebanyak 500. Lalu, alat diperintahkan untuk mengembalikan posisi *nozzle* seperti semula. Perintah tersebut diulang sebanyak 20 kali dengan setiap kali percobaan dikembalikan ke posisi semula walaupun terdapat selisih posisi awal dan akhir. Berikut merupakan hasil dari pengujian yang dilakukan di masing-masing sumbu.

Tabel 7: Hasil Pengujian Akurasi di Sumbu X

Percobaan ke-	Jarak Seharusnya (cm)	Jarak Asli (cm)	Selisih (cm)	Akurasi (%)
1	42,175	40,3	1,875	95,5
2	42,175	40,3	1,875	95,5
3	42,175	40,3	1,875	95,5
4	42,175	40,3	1,875	95,5
5	42,175	40,3	1,875	95,5
6	42,175	40,3	1,875	95,5

7	42,175	40,3	1,875	95,5
8	42,175	40,3	1,875	95,5
9	42,175	40,3	1,875	95,5
10	42,175	40,4	1,775	95,8
11	42,175	40,3	1,875	95,5
12	42,175	40,3	1,875	95,5
13	42,175	40,35	1,825	95,6
14	42,175	40,35	1,825	95,6
15	42,175	40,3	1,875	95,5
16	42,175	40,3	1,875	95,5
17	42,175	40,3	1,875	95,5
18	42,175	40,3	1,875	95,5
19	42,175	40,35	1,825	95,6
20	42,175	40,3	1,875	95,5
Rata-Rata Akurasi				95,53%
Repeatability				0,1 cm

Tabel 8: Hasil Pengujian Akurasi di Sumbu Y

Percobaan ke-	Jarak Seharusnya (cm)	Jarak Asli (cm)	Selisih (cm)	Akurasi (%)
1	10,3	10,2	0,1	99
2	10,3	10,3	0	100
3	10,3	10,2	0,1	99
4	10,3	10,2	0,1	99
5	10,3	10,3	0	100
6	10,3	10,3	0	100
7	10,3	10,2	0,1	99
8	10,3	10,2	0,1	99
9	10,3	10,2	0,1	99
10	10,3	10,2	0,1	99
11	10,3	10,2	0,1	99
12	10,3	10,2	0,1	99

13	10,3	10,1	0,2	98
14	10,3	10,2	0,1	99
15	10,3	10,3	0	100
16	10,3	10,3	0	100
17	10,3	10,2	0,1	99
18	10,3	10,2	0,1	99
19	10,3	10,1	0,2	98
20	10,3	10,2	0,1	99
Rata-Rata Akurasi				99,15%
Repeatability				0,2 cm

Dari *Tabel 7* dan *Tabel 8* di atas yang merupakan hasil pengujian akurasi di sumbu X dan sumbu Y, diperoleh rata-rata akurasi jarak tempuh *nozzle* berdasarkan *step* yang dimasukkan ke program. Akurasi diperoleh dari persentase jarak asli terhadap jarak seharusnya berdasarkan hasil perhitungan pengujian jarak. Nilai yang didapat adalah 95,53% untuk sumbu X dan 99,15% untuk sumbu Y. Nilai tersebut tergolong sangat baik karena masih di atas 95%. *Repeatability* diperoleh dari pengurangan selisih terbesar dengan selisih terkecil. Diperoleh *repeatability* dari pergerakan sumbu X dan sumbu Y secara berturut-turut sebesar 0,1 cm dan 0,2 cm. Nilai *repeatability* yang diperoleh tergolong sangat baik karena masih di bawah 1 cm.

Terdapat beberapa faktor yang memengaruhi tingkat akurasi dan *repeatability* pergerakan *nozzle*. Faktor yang pertama adalah kualitas bahan dari purwarupa. Bahan *rod steel* untuk jalur pergerakan *nozzle* yang digunakan harus lurus dan halus agar pergerakan dari *nozzle* selalu akurat. Faktor yang kedua adalah kualitas *hardware* yang digunakan, di mana tegangan dan arus yang dihasilkan dapat berpengaruh ke hasil pengujian. Faktor yang ketiga adalah akurasi pembangunan purwarupa yang berupa kerangka mesin CNC-nya. Pembangunan yang dilakukan harus rapi dan akurat dalam perhitungan dan penempatan masing-masing bahan dan komponen. Faktor yang terakhir adalah kecelakaan, di mana benturan dan pergeseran komponen akan berpengaruh sehingga dapat membuat hasil pengujian berbeda-beda di tiap pengujiannya.

4.4 Hasil Pengujian Waktu Proses Otomasi

Pengujian waktu proses otomasi telah dilakukan dengan cara memberikan perintah kepada alat robot agar bergerak ke koordinat yang ditentukan dan melakukan otomasi. Otomasi yang dilakukan dilakukan dalam beberapa kondisi, yaitu perlu menyiram tanaman 1 saja, perlu menyiram tanaman 2 saja, perlu menyiram tanaman 3 saja, perlu menyiram tanaman 1 & 2, perlu menyiram tanaman 1 & 3, perlu menyiram tanaman 2 & 3, dan perlu menyiram tanaman 1, 2, dan 3.

Posisi dari tanaman 1 adalah 1000 *step* sumbu X, posisi dari tanaman 2 adalah 2000 *step* sumbu X, dan posisi dari tanaman 3 adalah 3000 *step*.

Pengujian ini untuk membandingkan waktu yang ditempuh antara dua algoritma pemrograman dan penyiraman. Algoritma pertama adalah dengan membuat *void* terhadap masing-masing kondisi di mana jalur yang ditempuh berbeda-beda yang nantinya akan dipanggil tergantung data yang didapat. Algoritma kedua adalah dengan selalu menempuh jalur yang sama di kondisi manapun, tetapi hanya melakukan penyiraman terhadap tanaman yang kondisinya butuh disiram sesuai data yang didapat. Berikut merupakan hasil dari pengujian waktu proses otomasi.

Tabel 9: Hasil Waktu Tempuh Pergerakan Nozzle ke Target Tanaman

No	Tanaman Target	Waktu Tempuh Algoritma 1 (sekon)	Waktu Tempuh Algoritma 2 (sekon)
1	1	3,5	9,22
2	2	5,25	9,22
3	3	7,22	9,22
4	1 dan 2	6,16	9,22
5	1 dan 3	8,16	9,22
6	2 dan 3	8,2	9,22
7	1, 2, dan 3	9,22	9,22

Tabel 9 di atas merupakan hasil nilai waktu yang diperlukan untuk alat otomasi menggerakkan *nozzle* ke tanaman target. Waktu tersebut belum ditambah waktu *delay* program penyiraman dan waktu penyiraman berlangsung. Waktu jika ditambah penyiraman akan bertambah 4 detik. Berikut merupakan waktu tempuh otomasi penyiraman di masing-masing algoritma dan kondisi tanaman.

Tabel 10: Hasill Waktu Tempuh Otomasi Penyiraman

No	Tanaman Target	Waktu Tempuh Algoritma 1 (sekon)	Waktu Tempuh Algoritma 2 (sekon)
1	1	7,5	13,22
2	2	9,25	13,22
3	3	11,22	13,22
4	1 dan 2	14,16	17,22
5	1 dan 3	16,16	17,22
6	2 dan 3	16,2	17,22
7	1, 2, dan 3	21,22	21,22
Total		95,71	112,54

Hasil dari pengujian yang tertera pada *Tabel 10* menunjukkan bahwa penggunaan algoritma pertama lebih efisien dalam waktu karena melakukan otomasi sesuai dengan jalur yang perlu dilewati saja sedangkan algoritma kedua melakukan otomasi dengan jalur menghampiri semua tanaman tanpa peduli tanaman mana saja yang perlu disiram. Namun, algoritma kedua lebih efisien dalam kalimat pemrograman karena tidak butuh membuat fungsi *void* yang banyak sesuai kondisi yang mungkin terjadi. Apalagi, misal tanaman target ditambah menjadi 4 buah, maka akan ada total 14 *void* yang perlu dibuat. Penjelasan kelebihan dan kekurangan masing-masing algoritma dapat dilihat di *Tabel 11* berikut ini.

Tabel 11: Kelebihan & Kekurangan Antar Algoritma

Algoritma	Kelebihan	Kekurangan
1	Efisien dalam waktu dan daya	Tidak efisien dalam kode program
2	Efisien dalam kode program	Tidak efisien dalam waktu dan daya

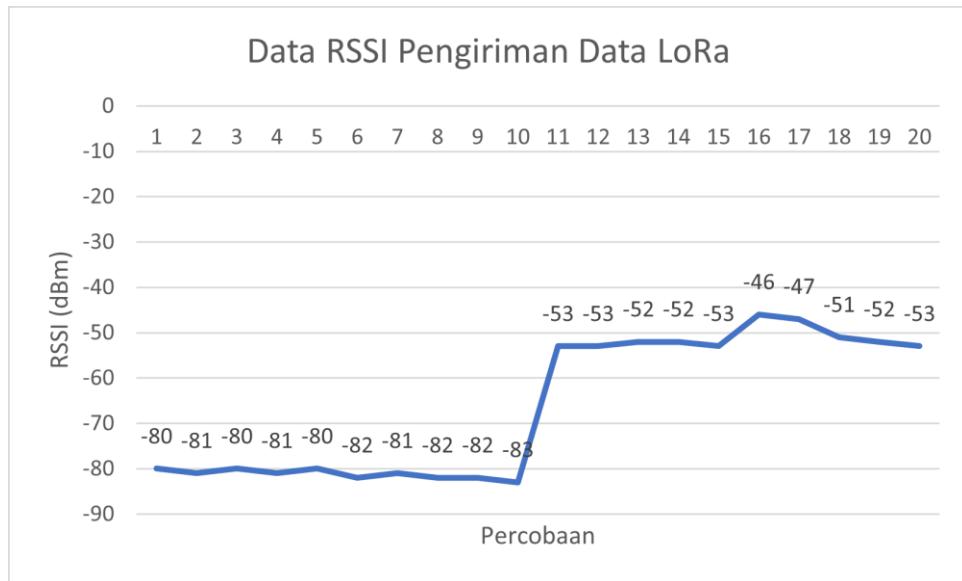
4.5 Hasil Pengukuran Parameter Pengiriman LoRa

Telah dilakukan pengukuran parameter pengiriman LoRa dengan mencatat nilai RSSI dan SNR dari *serial monitor* Arduino IDE. Nilai RSSI dan SNR diperoleh dari *syntax* yang telah disediakan oleh *library* “LoRalib.h”. Pengambilan data dilakukan sebanyak 20 kali dengan hasil sebagai berikut.

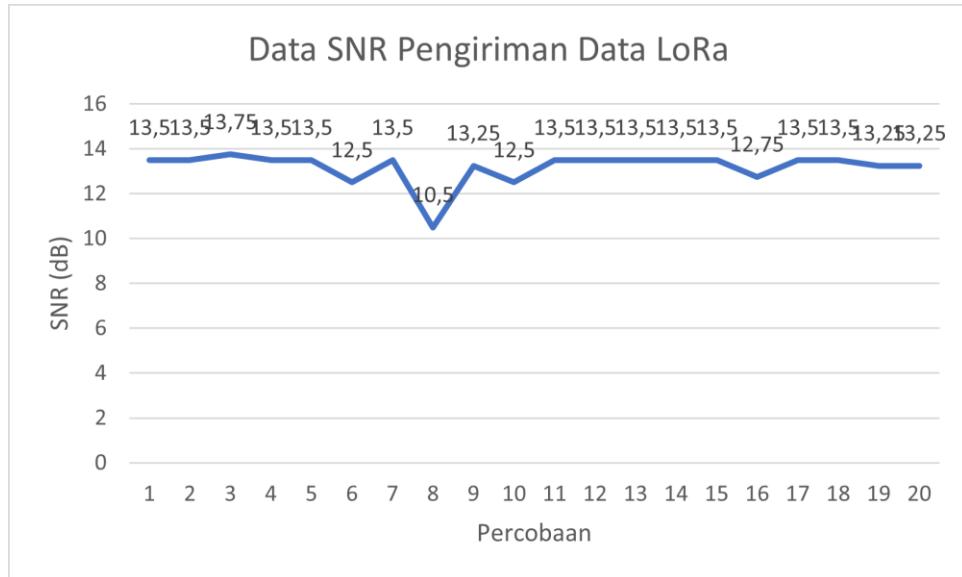
Tabel 12: Data Nilai RSSI dan SNR LoRa Receiver

Percobaan ke-	RSSI (dBm)	SNR (dB)
1	-80	13,5
2	-81	13,5
3	-80	13,75
4	-81	13,5
5	-80	13,5
6	-82	12,5
7	-81	13,5
8	-82	10,5
9	-82	13,25
10	-83	12,5
11	-53	13,5
12	-53	13,5
13	-52	13,5
14	-52	13,5
15	-53	13,5
16	-46	12,75
17	-47	13,5
18	-51	13,5
19	-52	13,25
20	-53	13,25
Rata-Rata	-66,2	13,1875

Tabel 11 di atas merupakan hasil dari pengambilan data RSSI dan SNR dari 20 kali percobaan yang dilakukan dan terdapat hasil nilai rata-rata RSSI dan SNR yang diperoleh. *Gambar 35* di bawah ini berisi grafik data RSSI yang diperoleh dan *Gambar 36* di atas berisi grafik data SNR yang diperoleh.



Gambar 35: Data RSSI Pengiriman Data LoRa



Gambar 36: Data SNR Pengiriman Data LoRa

Data 1 – 10 diperoleh saat kondisi di mana jarak dari LoRa *transmitter* dan LoRa *receiver* sekitar 2 m, sedangkan data 11 – 20 diperoleh saat kondisi di mana jarak dari LoRa *transmitter* dan LoRa *receiver* sekitar 15 cm. Oleh karena itu, dapat dikatakan bahwa jarak dapat memengaruhi nilai RSSI. Nilai SNR cenderung stabil di jarak berapapun karena faktor yang memengaruhinya adalah kualitas alat yang digunakan.

Berdasarkan hasil rekap data nilai RSSI dan SNR dari 20 kali transmisi data dilakukan, diperoleh rata-rata RSSI sebesar -66,2 dBm dan rata-rata SNR sebesar 13,1875 dB. RSSI masih tergolong baik karena masih dalam rentang -1 dBm hingga -99 dBm dan SNR tergolong baik karena berada di rentang 11,0 dB – 19,9 dB.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil pembangunan purwarupa, pengujian alat, dan analisis performansi pada tugas akhir ini, dapat ditarik kesimpulan sebagai berikut:

1. *Smart Greenbox* dengan purwarupa sistem otomasi berbasis IoT terhadap tanaman cabai telah berhasil dibangun.
2. Purwarupa sistem otomasi dapat melakukan penyiraman air atau pupuk cair kepada tanaman secara terprogram ataupun otomatis berdasarkan input dari sensor.
3. Akurasi rata-rata pergerakan *nozzle* mencapai nilai 95,53% untuk jalur sumbu X dan 99,15% untuk jalur sumbu Y.
4. Purwarupa sistem otomasi membutuhkan waktu total selama 95,71 sekon untuk melakukan penyiraman terhadap 7 kondisi tanaman yang berbeda dengan jumlah tanaman sebanyak 3 pot.
5. Performansi purwarupa sistem otomasi dipengaruhi oleh kualitas bahan, kualitas komponen, serta kerapian pembangunan.
6. Nilai RSSI dan SNR rata-rata transmisi data LoRa penelitian ini masuk ke kategori baik dengan nilai -66,2 dBm dan 13,1875 dB.
7. Penggunaan purwarupa sistem otomasi berpengaruh terhadap pertumbuhan tanaman karena parameter kelembaban tanah atau pH tanah selalu dalam keadaan optimal.
8. *Smart Greenbox* dengan purwarupa sistem otomasi cocok untuk digunakan di rumah karena bentuk yang minimalis dibandingkan dengan sistem *greenhouse* di mana cakupan tanaman dapat diperluas dan diperbanyak karena sistem otomasi dapat bergerak dengan sumbu X dan Y.

5.2 Saran

Dari penelitian yang telah dilakukan oleh penulis, terdapat beberapa saran untuk para pembaca yang dapat dijadikan bahan untuk pengembangan penelitian selanjutnya sebagai berikut:

1. Memastikan bahan yang digunakan tidak ada cacat agar proses otomasi berjalan dengan mulus.
2. Merangkai kerangka alat dan mesin CNC dengan akurat agar perhitungan jarak dapat lebih mudah dan benar.
3. Purwarupa sistem otomasi dapat ditambah *nozzle* tambahan atau menggunakan *nozzle* segitiga agar dapat melakukan otomasi penyiraman dan juga pemberian pupuk cair.
4. Pengujian dapat dilakukan dengan percobaan yang lebih banyak agar didapatkan tingkat akurasi yang lebih menjanjikan.

DAFTAR PUSTAKA

- [1] M. P. T. Sulistyanto, E. Febriawan, and I. Sulistiyowati, “Web of Things to control planting seeds and watering plants for indoor smart farm,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1098, no. 5, p. 052109, 2021, doi: 10.1088/1757-899x/1098/5/052109.
- [2] M. B. Eldion, “Rancang Bangun Sistem Sensor pada Otomasi Greenhouse Urban Farming,” 2017, [Online]. Available: <https://repository.its.ac.id/43818/>.
- [3] F. K. UMAH, “PENGARUH PEMBERIAN PUPUK HAYATI (BIOFERTILIZER) DAN MEDIA TANAM YANG BERBEDA PADA PERTUMBUHAN DAN PRODUKTIVITAS TANAMAN CABAI RAWIT (*Capsicum frutescens L.*) DI POLYBAG,” *Univ. Airlangga*, pp. 9–38, 2012.
- [4] S. R. Prathibha, A. Hongal, and M. P. Jyothi, “IOT Based Monitoring System in Smart Agriculture,” *Proc. - 2017 Int. Conf. Recent Adv. Electron. Commun. Technol. ICRAECT 2017*, pp. 81–84, 2017, doi: 10.1109/ICRAECT.2017.52.
- [5] W. Wilianto and A. Kurniawan, “Sejarah, Cara Kerja Dan Manfaat Internet of Things,” *Matrix J. Manaj. Teknol. dan Inform.*, vol. 8, no. 2, pp. 36–41, 2018, doi: 10.31940/matrix.v8i2.818.
- [6] I. A. Saputro, J. E. Suseno, and E. Widodo, “Rancang bangun sistem pengaturan kelembaban tanah secara real time menggunakan mikrokontroler dan diakses di web,” *Youngster Phys. J.*, vol. 6, no. 1, pp. 40–47, 2017.
- [7] L. C. Wijaya, “Analisis Usabilitas pada Sistem Monitoring dan Otomasi Greenhouse untuk Budidaya Tanaman Cabai Berbasis Android,” *Edu Komputika J.*, vol. 6, no. 2, pp. 60–67, 2019, doi: 10.15294/edukomputika.v6i2.34412.
- [8] H. L. Kushwaha, “Robotic and Mechatronic Application in Agriculture,” *RASSA J. Sci. Soc.*, vol. 1, no. 3, pp. 89–97, 2019.
- [9] M. Erick, S. Fiestas, R. Sixto, and G. Prado, “Modeling and Simulation of Kinematics and Trajectory Planning of a Farmbot Cartesian Robot,” *Proc. 2018 IEEE 25th Int. Conf. Electron. Electr. Eng. Comput. INTERCON 2018*, pp. 1–4, 2018, doi: 10.1109/INTERCON.2018.8526399.
- [10] A. Muchlis, W. Ridwan, and I. Z. Nasibu, “Rancang Bangun Mesin CNC (Computer Numerical Control) Laser dengan Metode Design for Assembly,” *Jambura J. Electr. Electron. Eng.*, vol. 3, no. 1, pp. 23–27, 2021, doi: 10.37905/jjeee.v3i1.9228.
- [11] M. S. Roni, “Analisis Parameter LoRa Pada Lingkungan Indoor,” *Repos. Univ. Din.*, vol., no., p. 8, 2020, [Online]. Available: repository.dinamika.ac.id.
- [12] Q. Zhou, K. Zheng, L. Hou, J. Xing, and R. Xu, “Design and implementation

- of open LORa for IoT,” *IEEE Access*, vol. 7, pp. 100649–100657, 2019, doi: 10.1109/ACCESS.2019.2930243.
- [13] P. Majda and B. Powalka, “Accuracy and repeatability positioning of high-performancel athe for non-circular turning,” *Arch. Mech. Technol. Mater.*, vol. 37, no. 1, pp. 85–90, 2017, doi: 10.1515/amtm-2017-0014.
 - [14] P. Studi, T. Elektro, F. Sains, D. A. N. Teknologi, and U. S. Dharma, “Aplikasi Robot Kartesian Pada Sistem Final Project Application of Cartesian Robot on,” 2018.
 - [15] N. Mungai Bryan, K. Fei Thang, and T. Vinesh, “An Urban Based Smart IOT Farming System,” *IOP Conf. Ser. Earth Environ. Sci.*, vol. 268, no. 1, 2019, doi: 10.1088/1755-1315/268/1/012038.
 - [16] A. B. Nugroho, M. A. Auliq, and M. Z. Alrasyid, “Analisa Perbandingan Performansi Akurasi Mesin CNC (Computer Numerical Control) Router Berbasis Mach3 dan Arduino Uno Menggunakan Metode SQC (Statistical Quality Control),” *J. Tek. Elektro dan Komputasi*, vol. 2, no. 2, pp. 75–86, 2020, doi: 10.32528/elkom.v2i2.3464.
 - [17] W. Jitviriya, N. Pudchuen, A. Phunopas, and E. Hayashi, “Design and modeling of an automatic cartesian farming robot,” *Proc. Int. Conf. Artif. Life Robot.*, vol. 2020, pp. 448–451, 2020, doi: 10.5954/ICAROB.2020.GS4-1.
 - [18] S. Kautsar, E. Rosdiana, B. Widiawan, D. P. S. Setyohadi, H. Y. Riskiawan, and R. Firgiyanto, “Farming Bot: Precision Agriculture System in Limited Land Based on Computer Numerical Control (CNC),” *IOP Conf. Ser. Earth Environ. Sci.*, vol. 411, no. 1, 2020, doi: 10.1088/1755-1315/411/1/012059.
 - [19] T. A. Zuraiyah, M. I. Suriansyah, and A. P. Akbar, “Smart Urban Farming Berbasis Internet Of Things (IoT),” *Inf. Manag. Educ. Prof.*, vol. 3, no. 2, pp. 139–150, 2019.
 - [20] S. Namani and B. Gonen, “Smart agriculture based on IoT and cloud computing,” *Proc. - 3rd Int. Conf. Inf. Comput. Technol. ICICT 2020*, pp. 553–556, 2020, doi: 10.1109/ICICT50521.2020.00094.
 - [21] T. Hidayat, “Internet of Things Smart Agriculture on ZigBee: A Systematic Review,” *J. Telekomun. dan Komput.*, vol. 8, no. 1, p. 75, 2017, doi: 10.22441/incomtech.v8i1.2146.
 - [22] H. Kim, *Studies in Computational Intelligence 930 Software Engineering in IoT , Big Data , Cloud and Mobile Computing*.
 - [23] Admin, “Arduino Uno R3” *Arduino.cc*, 2022, [Online]. Available: <https://docs.arduino.cc/hardware/uno-rev3>
 - [24] Admin, “Motor Stepper Nema 17 Extra Long Bipolar 4 Wire 3D Printer CNC” *depoinovasi.com*, 2011, [Online]. Available: <https://www.depoinovasi.com/produk-971-motor-stepper-nema-17-extra-long-bipolar-4-wire-3d-printer-cnc.html>

LAMPIRAN

1. Program Arduino IDE

a. Otomasi_Algoritma_1.ino

```
#include <LoRaLib.h>
#include <Servo.h>
```

```
#define dirPin 5
#define YdirPin 6
#define stepPin 2
#define YstepPin 3
```

```
#define RelayPin 7
```

```
SX1278 lora = new LoRa(10,17,16);
int Vrdata = 0;
```

```
String data;
int nilai_soil1, nilai_soil2, nilai_soil3;
float nilai_cahaya;
float suhu, kelembaban;
float pHtanah;
float pHtanah2;
float pHtanah3;
int timer;
```

```
int lamasiram = 1000;
```

```
void setup() {
```

```
    Serial.begin(9600);
    nilai_soil1 = 1000;
    nilai_soil2 = 1000;
    nilai_soil3 = 1000;
    // initialize SX1278 with default settings
    Serial.print(F("Initializing ... "));
    // carrier frequency:      434.0 MHz
    // bandwidth:             125.0 kHz
    // spreading factor:      9
    // coding rate:           7
    // sync word:              0x12
    // output power:          17 dBm
    // current limit:         100 mA
```

```

// preamble length:      8 symbols
// amplifier gain:      0 (automatic gain control)
int state = lora.begin();
if (state == ERR_NONE) {
    Serial.println(F("success!"));
} else {
    Serial.print(F("failed, code "));
    Serial.println(state);
    while (true);
}

// Declare pins as output:
pinMode(stepPin, OUTPUT);
pinMode(dirPin, OUTPUT);
pinMode(YstepPin, OUTPUT);
pinMode(YdirPin, OUTPUT);
pinMode(RelayPin, OUTPUT);
pinMode(8, OUTPUT);
digitalWrite(RelayPin, LOW);
//digitalWrite(8, LOW); // aktif
digitalWrite(8, HIGH); // non aktif
delay(5000);
}

void loop() {

//terima data
terimadata();

//otomasi
if ( nilai_soil1 < 250 && nilai_soil2 < 250 && nilai_soil3 < 250 ) {
    tanaman123();
    delay(5000);
}
else if ( nilai_soil1 < 250 && nilai_soil2 >= 250 && nilai_soil3 >= 250
) {
    tanaman1();
    delay(5000);
}
else if ( nilai_soil1 >= 250 && nilai_soil2 < 250 && nilai_soil3 >= 250
) {
    tanaman2();
    delay(5000);
}
}

```

```

else if ( nilai_soil1 >= 250 && nilai_soil2 >= 250 && nilai_soil3 < 250 )
{
    tanaman3();
    delay(5000);
}
else if ( nilai_soil1 < 250 && nilai_soil2 < 250 && nilai_soil3 >= 250 )
{
    tanaman12();
    delay(5000);
}
else if ( nilai_soil1 < 250 && nilai_soil2 >= 250 && nilai_soil3 < 250 )
{
    tanaman13();
    delay(5000);
}
else if ( nilai_soil1 >= 250 && nilai_soil2 < 250 && nilai_soil3 < 250 )
{
    tanaman23();
    delay(5000);
}
else {
    Serial.println("Tidak perlu disiram");
    delay(1000);
}

nilai_soil1 = 1000;
nilai_soil2 = 1000;
nilai_soil3 = 1000;
}

void terimadata() {

    Serial.print(F("Waiting for incoming transmission ... "));

    // you can receive data as an Arduino String
    // NOTE: receive() is a blocking method!
    //       See example ReceiveInterrupt for details
    //       on non-blocking reception method.
    // NOTE: for spreading factor 6, the packet length
    //       must be known in advance, and provided to
    //       receive() method!
    String str;
    int state = lora.receive(str);

    // you can also receive data as byte array
}

```

```

/*
size_t len = 8;
byte byteArr[len];
int state = lora.receive(byteArr, len);
*/

if (state == ERR_NONE) {
    // packet was successfully received
    Serial.println(F("success!"));

    // print data of the packet
    Serial.print(F("Data:\t\t\t"));
    Serial.println(str);

    // print RSSI (Received Signal Strength Indicator)
    // of the last received packet
    Serial.print(F("RSSI:\t\t\t"));
    Serial.print(lora.getRSSI());
    Serial.println(F(" dBm"));

    // print SNR (Signal-to-Noise Ratio)
    // of the last received packet
    Serial.print(F("SNR:\t\t\t"));
    Serial.print(lora.getSNR());
    Serial.println(F(" dB"));

    // print frequency error
    // of the last received packet
    Serial.print(F("Frequency error:\t"));
    Serial.print(lora.getFrequencyError());
    Serial.println(F(" Hz"));

    data = str;

    String data_nilai0 = getValue(data, ',', 0);
    String data_nilai1 = getValue(data, ',', 1);
    String data_nilai2 = getValue(data, ',', 2);
    String data_nilai3 = getValue(data, ',', 3);
    String data_nilai4 = getValue(data, ',', 4);
    String data_nilai5 = getValue(data, ',', 5);
    String data_nilai6 = getValue(data, ',', 6);
    String data_nilai7 = getValue(data, ',', 7);
    String data_nilai8 = getValue(data, ',', 8);

    nilai_soil1 = data_nilai0.toInt();
}

```

```

nilai_soil2 = data_nilai1.toInt();
nilai_soil3 = data_nilai2.toInt();
nilai_cahaya = data_nilai3.toFloat();
suhu = data_nilai4.toFloat();
kelembaban = data_nilai5.toFloat();
phTanah = data_nilai6.toFloat();
phTanah2 = data_nilai7.toFloat();
phTanah3 = data_nilai8.toFloat();

Serial.print("Soil1: ");
Serial.print(nilai_soil1);
Serial.print(" Soil2: ");
Serial.print(nilai_soil2);
Serial.print(" Soil3: ");
Serial.println(nilai_soil3);

Serial.print("Nilai Cahaya: ");
Serial.print(nilai_cahaya);
Serial.println(" lux");

Serial.print("kelembaban: ");
Serial.print(kelembaban);
Serial.print(" ");
Serial.print("suhu: ");
Serial.println(suhu);

Serial.print("pH Tanah= ");
Serial.println(phTanah);
Serial.print("pH Tanah2= ");
Serial.println(phTanah2);
Serial.print("pH Tanah3= ");
Serial.println(phTanah3);

timer++;
Serial.print("Timer: ");
Serial.println(timer);

Serial.println();

delay(100);

} else if (state == ERR_RX_TIMEOUT) {
// timeout occurred while waiting for a packet

```

```

Serial.println(F("timeout!"));

} else if (state == ERR_CRC_MISMATCH) {
    // packet was received, but is malformed
    Serial.println(F("CRC error!"));

}

}

String getValue(String data, char separator, int index) {
    int found = 0;
    int strIndex[] = { 0, -1 };
    int maxIndex = data.length() - 1;
    for (int i = 0; i <= maxIndex && found <= index; i++) {
        if (data.charAt(i) == separator || i == maxIndex) {
            found++;
            strIndex[0] = strIndex[1] + 1;
            strIndex[1] = (i == maxIndex) ? i + 1 : i;
        }
    }
    return found > index ? data.substring(strIndex[0], strIndex[1]) : "";
}

void siram() {
    delay(2000);
    digitalWrite(RelayPin, HIGH);
    delay(lamasiram);
    digitalWrite(RelayPin, LOW);
}

void tanaman1() {
    Serial.println("Menyiram tanaman 1...");
    delay(2000);
    digitalWrite(8, LOW); // aktif
    digitalWrite(dirPin, HIGH);
    for (int i = 0; i < 1000; i++) {
        // These four lines result in 1 step:
        digitalWrite(stepPin, HIGH);
        delayMicroseconds(500);
        digitalWrite(stepPin, LOW);
        delayMicroseconds(500);
    }
    digitalWrite(8, HIGH); // non aktif
    delay(1000);
}

```

```

siram();
delay(1000);
digitalWrite(8, LOW); // aktif
digitalWrite(dirPin, LOW);
for (int i = 0; i < 1000; i++) {
    // These four lines result in 1 step:
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(500);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(500);
}
digitalWrite(8, HIGH); // non aktif
delay(1000);
}

void tanaman2() {
    Serial.println("Menyiram tanaman 2...\"");
    delay(2000);
    digitalWrite(8, LOW); // aktif
    digitalWrite(dirPin, HIGH);
    for (int i = 0; i < 2000; i++) {
        // These four lines result in 1 step:
        digitalWrite(stepPin, HIGH);
        delayMicroseconds(500);
        digitalWrite(stepPin, LOW);
        delayMicroseconds(500);
    }
    digitalWrite(8, HIGH); // non aktif
    delay(1000);
    siram();
    delay(1000);
    digitalWrite(8, LOW); // aktif
    digitalWrite(dirPin, LOW);
    for (int i = 0; i < 2000; i++) {
        // These four lines result in 1 step:
        digitalWrite(stepPin, HIGH);
        delayMicroseconds(500);
        digitalWrite(stepPin, LOW);
        delayMicroseconds(500);
    }
    digitalWrite(8, HIGH); // non aktif
    delay(1000);
}

void tanaman3() {

```

```

Serial.println("Menyiram tanaman 3...");
delay(2000);
digitalWrite(8, LOW); // aktif
digitalWrite(dirPin, HIGH);
for (int i = 0; i < 3000; i++) {
    // These four lines result in 1 step:
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(500);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(500);
}
digitalWrite(8, HIGH); // non aktif
delay(1000);
siram();
delay(1000);
digitalWrite(8, LOW); // aktif
digitalWrite(dirPin, LOW);
for (int i = 0; i < 3000; i++) {
    // These four lines result in 1 step:
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(500);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(500);
}
digitalWrite(8, HIGH); // non aktif
delay(1000);
}

void tanaman12() {
    Serial.println("Menyiram tanaman 1 & 2...");
    delay(2000);
    digitalWrite(8, LOW); // aktif
    digitalWrite(dirPin, HIGH);
    for (int i = 0; i < 1000; i++) {
        // These four lines result in 1 step:
        digitalWrite(stepPin, HIGH);
        delayMicroseconds(500);
        digitalWrite(stepPin, LOW);
        delayMicroseconds(500);
    }
    digitalWrite(8, HIGH); // non aktif
    delay(1000);
    siram();
    delay(1000);
    digitalWrite(8, LOW); // aktif
}

```

```

digitalWrite(dirPin, HIGH);
for (int i = 0; i < 1000; i++) {
    // These four lines result in 1 step:
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(500);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(500);
}
digitalWrite(8, HIGH); // non aktif
delay(1000);
siram();
delay(1000);
digitalWrite(8, LOW); // aktif
digitalWrite(dirPin, LOW);
for (int i = 0; i < 2000; i++) {
    // These four lines result in 1 step:
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(500);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(500);
}
digitalWrite(8, HIGH); // non aktif
delay(1000);
}

void tanaman13() {
    Serial.println("Menyiram tanaman 1 & 3...");
    delay(2000);
    digitalWrite(8, LOW); // aktif
    digitalWrite(dirPin, HIGH);
    for (int i = 0; i < 1000; i++) {
        // These four lines result in 1 step:
        digitalWrite(stepPin, HIGH);
        delayMicroseconds(500);
        digitalWrite(stepPin, LOW);
        delayMicroseconds(500);
    }
    digitalWrite(8, HIGH); // non aktif
    delay(1000);
    siram();
    delay(1000);
    digitalWrite(8, LOW); // aktif
    digitalWrite(dirPin, HIGH);
    for (int i = 0; i < 2000; i++) {
        // These four lines result in 1 step:
    }
}

```

```

digitalWrite(stepPin, HIGH);
delayMicroseconds(500);
digitalWrite(stepPin, LOW);
delayMicroseconds(500);
}
digitalWrite(8, HIGH); // non aktif
delay(1000);
siram();
delay(1000);
digitalWrite(8, LOW); // aktif
digitalWrite(dirPin, LOW);
for (int i = 0; i < 3000; i++) {
// These four lines result in 1 step:
digitalWrite(stepPin, HIGH);
delayMicroseconds(500);
digitalWrite(stepPin, LOW);
delayMicroseconds(500);
}
digitalWrite(8, HIGH); // non aktif
delay(1000);
}

void tanaman23() {
Serial.println("Menyiram tanaman 2 & 3...");
delay(2000);
digitalWrite(8, LOW); // aktif
digitalWrite(dirPin, HIGH);
for (int i = 0; i < 2000; i++) {
// These four lines result in 1 step:
digitalWrite(stepPin, HIGH);
delayMicroseconds(500);
digitalWrite(stepPin, LOW);
delayMicroseconds(500);
}
digitalWrite(8, HIGH); // non aktif
delay(1000);
siram();
delay(1000);
digitalWrite(8, LOW); // aktif
digitalWrite(dirPin, HIGH);
for (int i = 0; i < 1000; i++) {
// These four lines result in 1 step:
digitalWrite(stepPin, HIGH);
delayMicroseconds(500);
digitalWrite(stepPin, LOW);
}
}

```

```

delayMicroseconds(500);
}
digitalWrite(8, HIGH); // non aktif
delay(1000);
siram();
delay(1000);
digitalWrite(8, LOW); // aktif
digitalWrite(dirPin, LOW);
for (int i = 0; i < 3000; i++) {
// These four lines result in 1 step:
digitalWrite(stepPin, HIGH);
delayMicroseconds(500);
digitalWrite(stepPin, LOW);
delayMicroseconds(500);
}
digitalWrite(8, HIGH); // non aktif
delay(1000);
}

void tanaman123() {
Serial.println("Menyiram tanaman 1, 2, dan 3...");
delay(2000);
digitalWrite(8, LOW); // aktif
digitalWrite(dirPin, HIGH);
for (int i = 0; i < 1000; i++) {
// These four lines result in 1 step:
digitalWrite(stepPin, HIGH);
delayMicroseconds(500);
digitalWrite(stepPin, LOW);
delayMicroseconds(500);
}
digitalWrite(8, HIGH); // non aktif
delay(1000);
siram();
delay(1000);
digitalWrite(8, LOW); // aktif
digitalWrite(dirPin, HIGH);
for (int i = 0; i < 1000; i++) {
// These four lines result in 1 step:
digitalWrite(stepPin, HIGH);
delayMicroseconds(500);
digitalWrite(stepPin, LOW);
delayMicroseconds(500);
}
digitalWrite(8, HIGH); // non aktif
}

```

```

delay(1000);
siram();
delay(1000);
digitalWrite(8, LOW); // aktif
digitalWrite(dirPin, HIGH);
for (int i = 0; i < 1000; i++) {
    // These four lines result in 1 step:
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(500);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(500);
}
digitalWrite(8, HIGH); // non aktif
delay(1000);
siram();
delay(1000);
digitalWrite(8, LOW); // aktif
digitalWrite(dirPin, LOW);
for (int i = 0; i < 3000; i++) {
    // These four lines result in 1 step:
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(500);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(500);
}
digitalWrite(8, HIGH); // non aktif
delay(1000);
}
b. Otomasi_Algoritma_2.ino
#include <LoRaLib.h>
#include <Servo.h>

#define dirPin 5
#define YdirPin 6
#define stepPin 2
#define YstepPin 3

#define RelayPin 7

SX1278 lora = new LoRa(10,17,16);
int Vrdata = 0;

String data;
int nilai_soil1, nilai_soil2, nilai_soil3;
float nilai_cahaya;

```

```

float suhu, kelembaban;
float pHanah;
float pHanah2;
float pHanah3;
int timer;

int lamasiram = 1000;

void setup() {

    Serial.begin(9600);
    nilai_soil1 = 1000;
    nilai_soil2 = 1000;
    nilai_soil3 = 1000;
    // initialize SX1278 with default settings
    Serial.print(F("Initializing ... "));
    // carrier frequency:        434.0 MHz
    // bandwidth:                125.0 kHz
    // spreading factor:         9
    // coding rate:              7
    // sync word:                0x12
    // output power:             17 dBm
    // current limit:            100 mA
    // preamble length:          8 symbols
    // amplifier gain:           0 (automatic gain control)
    int state = lora.begin();
    if (state == ERR_NONE) {
        Serial.println(F("success!"));
    } else {
        Serial.print(F("failed, code "));
        Serial.println(state);
        while (true);
    }

    // Declare pins as output:
    pinMode(stepPin, OUTPUT);
    pinMode(dirPin, OUTPUT);
    pinMode(YstepPin, OUTPUT);
    pinMode(YdirPin, OUTPUT);
    pinMode(RelayPin, OUTPUT);
    pinMode(8, OUTPUT);
    digitalWrite(RelayPin, LOW);
    //digitalWrite(8, LOW); // aktif
    digitalWrite(8, HIGH); // non aktif
    delay(5000);
}

```

```

}

void loop() {

    //terima data
    terimadata();

    //otomasi
    if ( nilai_soil1 < 250 || nilai_soil2 < 250 || nilai_soil3 < 250 ) {
        Serial.println("Melakukan otomasi...");
        otomasi();
    }
    else {
        Serial.println("Tidak perlu disiram");
    }

    nilai_soil1 = 1000;
    nilai_soil2 = 1000;
    nilai_soil3 = 1000;
}

void terimadata() {

    Serial.print(F("Waiting for incoming transmission ... "));

    // you can receive data as an Arduino String
    // NOTE: receive() is a blocking method!
    //       See example ReceiveInterrupt for details
    //       on non-blocking reception method.
    // NOTE: for spreading factor 6, the packet length
    //       must be known in advance, and provided to
    //       receive() method!
    String str;
    int state = lora.receive(str);

    // you can also receive data as byte array
    /*
     size_t len = 8;
     byte byteArr[len];
     int state = lora.receive(byteArr, len);
    */

    if (state == ERR_NONE) {
        // packet was successfully received
        Serial.println(F("success!"));
    }
}

```

```

// print data of the packet
Serial.print(F("Data:\t\t\t"));
Serial.println(str);

// print RSSI (Received Signal Strength Indicator)
// of the last received packet
Serial.print(F("RSSI:\t\t\t"));
Serial.print(lora.getRSSI());
Serial.println(F(" dBm"));

// print SNR (Signal-to-Noise Ratio)
// of the last received packet
Serial.print(F("SNR:\t\t\t"));
Serial.print(lora.getSNR());
Serial.println(F(" dB"));

// print frequency error
// of the last received packet
Serial.print(F("Frequency error:\t\t"));
Serial.print(lora.getFrequencyError());
Serial.println(F(" Hz"));

data = str;

String data_nilai0 = getValue(data, ',', 0);
String data_nilai1 = getValue(data, ',', 1);
String data_nilai2 = getValue(data, ',', 2);
String data_nilai3 = getValue(data, ',', 3);
String data_nilai4 = getValue(data, ',', 4);
String data_nilai5 = getValue(data, ',', 5);
String data_nilai6 = getValue(data, ',', 6);
String data_nilai7 = getValue(data, ',', 7);
String data_nilai8 = getValue(data, ',', 8);

nilai_soil1 = data_nilai0.toInt();
nilai_soil2 = data_nilai1.toInt();
nilai_soil3 = data_nilai2.toInt();
nilai_cahaya = data_nilai3.toFloat();
suhu = data_nilai4.toFloat();
kelembaban = data_nilai5.toFloat();
phTanah = data_nilai6.toFloat();
phTanah2 = data_nilai7.toFloat();
phTanah3 = data_nilai8.toFloat();

```

```

Serial.print("Soil1: ");
Serial.print(nilai_soil1);
Serial.print(" Soil2: ");
Serial.print(nilai_soil2);
Serial.print(" Soil3: ");
Serial.println(nilai_soil3);

Serial.print("Nilai Cahaya: ");
Serial.print(nilai_cahaya);
Serial.println(" lux");

Serial.print("kelembaban: ");
Serial.print(kelembaban);
Serial.print(" ");
Serial.print("suhu: ");
Serial.println(suhu);

Serial.print("pH Tanah= ");
Serial.println(phTanah);
Serial.print("pH Tanah2= ");
Serial.println(phTanah2);
Serial.print("pH Tanah3= ");
Serial.println(phTanah3);

timer++;
Serial.print("Timer: ");
Serial.println(timer);

Serial.println();

delay(100);

} else if (state == ERR_RX_TIMEOUT) {
// timeout occurred while waiting for a packet
Serial.println(F("timeout!"));

} else if (state == ERR_CRC_MISMATCH) {
// packet was received, but is malformed
Serial.println(F("CRC error!"));

}

}

```

```

String getValue(String data, char separator, int index) {
    int found = 0;
    int strIndex[] = { 0, -1 };
    int maxIndex = data.length() - 1;
    for (int i = 0; i <= maxIndex && found <= index; i++) {
        if (data.charAt(i) == separator || i == maxIndex) {
            found++;
            strIndex[0] = strIndex[1] + 1;
            strIndex[1] = (i == maxIndex) ? i + 1 : i;
        }
    }
    return found > index ? data.substring(strIndex[0], strIndex[1]) : "";
}

void siram() {
    delay(2000);
    digitalWrite(RelayPin, HIGH);
    delay(lamasiram);
    digitalWrite(RelayPin, LOW);
    delay(1000);
}

void otomasi() {
    delay(2000);

    //tanaman 1
    digitalWrite(8, LOW); // aktif
    digitalWrite(dirPin, HIGH);
    for (int i = 0; i < 1000; i++) {
        // These four lines result in 1 step:
        digitalWrite(stepPin, HIGH);
        delayMicroseconds(500);
        digitalWrite(stepPin, LOW);
        delayMicroseconds(500);
    }
    digitalWrite(8, HIGH); // non aktif
    delay(1000);
    if (nilai_soil1 < 250) {
        siram()
    }

    //tanaman 2
    digitalWrite(8, LOW); // aktif
    digitalWrite(dirPin, HIGH);
}

```

```

for (int i = 0; i < 1000; i++) {
    // These four lines result in 1 step:
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(500);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(500);
}
digitalWrite(8, HIGH); // non aktif
delay(1000);
if (nilai_soil2 < 250) {
    siram()
}

//tanaman 3
digitalWrite(8, LOW); // aktif
digitalWrite(dirPin, HIGH);
for (int i = 0; i < 1000; i++) {
    // These four lines result in 1 step:
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(500);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(500);
}
digitalWrite(8, HIGH); // non aktif
delay(1000);
if (nilai_soil3 < 250) {
    siram()
}

digitalWrite(8, LOW); // aktif
digitalWrite(dirPin, LOW);
for (int i = 0; i < 3000; i++) {
    // These four lines result in 1 step:
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(500);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(500);
}
digitalWrite(8, HIGH); // non aktif
delay(1000);
}

c. Tes_Nyiram.ino
#include <Servo.h>

#define RelayPin 7

```

```

int lamasiram = 1000;

void setup() {
    pinMode(RelayPin, OUTPUT);
    delay(5000);
}

void loop() {
    siram();
}

void siram() {
    digitalWrite(RelayPin, HIGH);
    delay(lamasiram);
    digitalWrite(RelayPin, LOW);
    delay(2000);
}

d. Uji_Mekanik.ino
#include <Servo.h>

#define dirPin 5
#define YdirPin 6
#define stepPin 2
#define YstepPin 3
#define stepsPerRevolution 2000
#define YstepsPerRevolution 1000
#define RelayPin 7

int steptest = 500;
int lamasiram = 1000;

void setup() {

    pinMode(stepPin, OUTPUT);
    pinMode(dirPin, OUTPUT);
    pinMode(YstepPin, OUTPUT);
    pinMode(YdirPin, OUTPUT);
    pinMode(RelayPin, OUTPUT);
    pinMode(8, OUTPUT);
    digitalWrite(RelayPin, LOW);
    //digitalWrite(8, LOW); // aktif
    digitalWrite(8, HIGH); // non aktif
    delay(5000);
}

```

```

}

void loop() {

    //ujimekanik();
    //tesjarak();
    //tesxpos();
    //tesxmin();
    //tesypos();
    //tesymin();

    //algo1();
    //algo2();
    //algo2();
    //algo2();
    //algo2();
    //algo2();
    //algo2();
    //algo2();
    //algo2();

}

void algo1() {
    tanaman1();
    tanaman2();
    tanaman3();
    tanaman12();
    tanaman13();
    tanaman23();
    tanaman123();
}

void algo2() {
    delay(2000);

    //tanaman 1
    digitalWrite(8, LOW); // aktif
    digitalWrite(dirPin, HIGH);
    for (int i = 0; i < 1000; i++) {
        // These four lines result in 1 step:
        digitalWrite(stepPin, HIGH);
        delayMicroseconds(500);
        digitalWrite(stepPin, LOW);
        delayMicroseconds(500);
    }
}

```

```

digitalWrite(8, HIGH); // non aktif
delay(1000);

//tanaman 2
digitalWrite(8, LOW); // aktif
digitalWrite(dirPin, HIGH);
for (int i = 0; i < 1000; i++) {
    // These four lines result in 1 step:
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(500);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(500);
}
digitalWrite(8, HIGH); // non aktif
delay(1000);

//tanaman 3
digitalWrite(8, LOW); // aktif
digitalWrite(dirPin, HIGH);
for (int i = 0; i < 1000; i++) {
    // These four lines result in 1 step:
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(500);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(500);
}
digitalWrite(8, HIGH); // non aktif
delay(1000);

digitalWrite(8, LOW); // aktif
digitalWrite(dirPin, LOW);
for (int i = 0; i < 3000; i++) {
    // These four lines result in 1 step:
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(500);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(500);
}
digitalWrite(8, HIGH); // non aktif
delay(1000);
}

void tesjarak() {
    delay(2000);
    digitalWrite(8, LOW); // aktif

```

```

digitalWrite(YdirPin, HIGH);
for (int i = 0; i < steptest; i++) {
    // These four lines result in 1 step:
    digitalWrite(YstepPin, HIGH);
    delayMicroseconds(500);
    digitalWrite(YstepPin, LOW);
    delayMicroseconds(500);
}
digitalWrite(8, HIGH); // non aktif
delay(15000);
digitalWrite(8, LOW); // aktif
digitalWrite(YdirPin, LOW);
for (int i = 0; i < steptest; i++) {
    // These four lines result in 1 step:
    digitalWrite(YstepPin, HIGH);
    delayMicroseconds(500);
    digitalWrite(YstepPin, LOW);
    delayMicroseconds(500);
}
digitalWrite(8, HIGH); // non aktif
delay(10000);
}

void tesxpos() {
    digitalWrite(8, LOW); // aktif
    digitalWrite(dirPin, HIGH);
    for (int i = 0; i < stepsPerRevolution; i++) {
        // These four lines result in 1 step:
        digitalWrite(stepPin, HIGH);
        delayMicroseconds(500);
        digitalWrite(stepPin, LOW);
        delayMicroseconds(500);
    }
    digitalWrite(8, HIGH); // non aktif
    delay(1000);
}

void tesxmin() {
    digitalWrite(8, LOW); // aktif
    digitalWrite(dirPin, LOW);
    for (int i = 0; i < stepsPerRevolution; i++) {
        // These four lines result in 1 step:
        digitalWrite(stepPin, HIGH);
        delayMicroseconds(500);
        digitalWrite(stepPin, LOW);
    }
}

```

```

        delayMicroseconds(500);
    }
    digitalWrite(8, HIGH); // non aktif
    delay(1000);
}

void tesypos() {
    digitalWrite(8, LOW); // aktif
    digitalWrite(YdirPin, LOW);
    for (int i = 0; i < YstepsPerRevolution; i++) {
        // These four lines result in 1 step:
        digitalWrite(YstepPin, HIGH);
        delayMicroseconds(500);
        digitalWrite(YstepPin, LOW);
        delayMicroseconds(500);
    }
    digitalWrite(8, HIGH); // non aktif
    delay(1000);
}

void tesymin() {
    digitalWrite(8, LOW); // aktif
    digitalWrite(YdirPin, HIGH);
    for (int i = 0; i < YstepsPerRevolution; i++) {
        // These four lines result in 1 step:
        digitalWrite(YstepPin, HIGH);
        delayMicroseconds(500);
        digitalWrite(YstepPin, LOW);
        delayMicroseconds(500);
    }
    digitalWrite(8, HIGH); // non aktif
    delay(1000);
}

void ujimekanik() {

    siram();
    ybawah();
    siram();
    yatas();

    xkanan();
    siram();
    ybawah();
    siram();
}

```

```

yatas();

xkanan();
siram();
ybawah();
siram();
yatas();

xkanan();
siram();
ybawah();
siram();
yatas();

xkanan();
siram();
ybawah();
siram();
yatas();

xkiri();

delay(100000);

}

void siram() {
    digitalWrite(RelayPin, HIGH);
    delay(lamasiram);
    digitalWrite(RelayPin, LOW);
}

void ybawah() {
    digitalWrite(8, LOW); // aktif
    digitalWrite(YdirPin, HIGH);
    for (int i = 0; i < YstepsPerRevolution; i++) {
        // These four lines result in 1 step:
        digitalWrite(YstepPin, HIGH);
        delayMicroseconds(500);
        digitalWrite(YstepPin, LOW);
        delayMicroseconds(500);
    }
    digitalWrite(8, HIGH); // non aktif
    delay(1000);
}

```

```

}

void yatas() {
    digitalWrite(8, LOW); // aktif
    digitalWrite(YdirPin, LOW);
    for (int i = 0; i < YstepsPerRevolution; i++) {
        // These four lines result in 1 step:
        digitalWrite(YstepPin, HIGH);
        delayMicroseconds(500);
        digitalWrite(YstepPin, LOW);
        delayMicroseconds(500);
    }
    digitalWrite(8, HIGH); // non aktif
    delay(1000);
}

void xkanan() {
    digitalWrite(8, LOW); // aktif
    digitalWrite(dirPin, HIGH);
    for (int i = 0; i < stepsPerRevolution; i++) {
        // These four lines result in 1 step:
        digitalWrite(stepPin, HIGH);
        delayMicroseconds(500);
        digitalWrite(stepPin, LOW);
        delayMicroseconds(500);
    }
    digitalWrite(8, HIGH); // non aktif
    delay(1000);
}

void xkiri() {
    digitalWrite(8, LOW); // aktif
    digitalWrite(dirPin, LOW);
    for (int i = 0; i < 4000; i++) {
        // These four lines result in 1 step:
        digitalWrite(stepPin, HIGH);
        delayMicroseconds(500);
        digitalWrite(stepPin, LOW);
        delayMicroseconds(500);
    }
    digitalWrite(8, HIGH); // non aktif
    delay(1000);
}

void tanaman1() {

```

```

Serial.println("Menyiram tanaman 1...");
delay(2000);
digitalWrite(8, LOW); // aktif
digitalWrite(dirPin, HIGH);
for (int i = 0; i < 1000; i++) {
    // These four lines result in 1 step:
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(500);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(500);
}
digitalWrite(8, HIGH); // non aktif
delay(1000);
digitalWrite(8, LOW); // aktif
digitalWrite(dirPin, LOW);
for (int i = 0; i < 1000; i++) {
    // These four lines result in 1 step:
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(500);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(500);
}
digitalWrite(8, HIGH); // non aktif
delay(1000);
}

void tanaman2() {
    Serial.println("Menyiram tanaman 2...");
    delay(2000);
    digitalWrite(8, LOW); // aktif
    digitalWrite(dirPin, HIGH);
    for (int i = 0; i < 2000; i++) {
        // These four lines result in 1 step:
        digitalWrite(stepPin, HIGH);
        delayMicroseconds(500);
        digitalWrite(stepPin, LOW);
        delayMicroseconds(500);
    }
    digitalWrite(8, HIGH); // non aktif
    delay(1000);
    digitalWrite(8, LOW); // aktif
    digitalWrite(dirPin, LOW);
    for (int i = 0; i < 2000; i++) {
        // These four lines result in 1 step:
        digitalWrite(stepPin, HIGH);

```

```

delayMicroseconds(500);
digitalWrite(stepPin, LOW);
delayMicroseconds(500);
}
digitalWrite(8, HIGH); // non aktif
delay(1000);
}

void tanaman3() {
Serial.println("Menyiram tanaman 3...");
delay(2000);
digitalWrite(8, LOW); // aktif
digitalWrite(dirPin, HIGH);
for (int i = 0; i < 3000; i++) {
// These four lines result in 1 step:
digitalWrite(stepPin, HIGH);
delayMicroseconds(500);
digitalWrite(stepPin, LOW);
delayMicroseconds(500);
}
digitalWrite(8, HIGH); // non aktif
delay(1000);
digitalWrite(8, LOW); // aktif
digitalWrite(dirPin, LOW);
for (int i = 0; i < 3000; i++) {
// These four lines result in 1 step:
digitalWrite(stepPin, HIGH);
delayMicroseconds(500);
digitalWrite(stepPin, LOW);
delayMicroseconds(500);
}
digitalWrite(8, HIGH); // non aktif
delay(1000);
}

void tanaman12() {
Serial.println("Menyiram tanaman 1 & 2...");
delay(2000);
digitalWrite(8, LOW); // aktif
digitalWrite(dirPin, HIGH);
for (int i = 0; i < 1000; i++) {
// These four lines result in 1 step:
digitalWrite(stepPin, HIGH);
delayMicroseconds(500);
digitalWrite(stepPin, LOW);
}
}

```

```

delayMicroseconds(500);
}
digitalWrite(8, HIGH); // non aktif
delay(1000);
digitalWrite(8, LOW); // aktif
digitalWrite(dirPin, HIGH);
for (int i = 0; i < 1000; i++) {
    // These four lines result in 1 step:
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(500);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(500);
}
digitalWrite(8, HIGH); // non aktif
delay(1000);
digitalWrite(8, LOW); // aktif
digitalWrite(dirPin, LOW);
for (int i = 0; i < 2000; i++) {
    // These four lines result in 1 step:
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(500);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(500);
}
digitalWrite(8, HIGH); // non aktif
delay(1000);
}

void tanaman13() {
    Serial.println("Menyiram tanaman 1 & 3...");
    delay(2000);
    digitalWrite(8, LOW); // aktif
    digitalWrite(dirPin, HIGH);
    for (int i = 0; i < 1000; i++) {
        // These four lines result in 1 step:
        digitalWrite(stepPin, HIGH);
        delayMicroseconds(500);
        digitalWrite(stepPin, LOW);
        delayMicroseconds(500);
    }
    digitalWrite(8, HIGH); // non aktif
    delay(1000);
    digitalWrite(8, LOW); // aktif
    digitalWrite(dirPin, HIGH);
    for (int i = 0; i < 2000; i++) {

```

```

// These four lines result in 1 step:
digitalWrite(stepPin, HIGH);
delayMicroseconds(500);
digitalWrite(stepPin, LOW);
delayMicroseconds(500);
}
digitalWrite(8, HIGH); // non aktif
delay(1000);
digitalWrite(8, LOW); // aktif
digitalWrite(dirPin, LOW);
for (int i = 0; i < 3000; i++) {
// These four lines result in 1 step:
digitalWrite(stepPin, HIGH);
delayMicroseconds(500);
digitalWrite(stepPin, LOW);
delayMicroseconds(500);
}
digitalWrite(8, HIGH); // non aktif
delay(1000);
}

void tanaman23() {
Serial.println("Menyiram tanaman 2 & 3...");
delay(2000);
digitalWrite(8, LOW); // aktif
digitalWrite(dirPin, HIGH);
for (int i = 0; i < 2000; i++) {
// These four lines result in 1 step:
digitalWrite(stepPin, HIGH);
delayMicroseconds(500);
digitalWrite(stepPin, LOW);
delayMicroseconds(500);
}
digitalWrite(8, HIGH); // non aktif
delay(1000);
digitalWrite(8, LOW); // aktif
digitalWrite(dirPin, HIGH);
for (int i = 0; i < 1000; i++) {
// These four lines result in 1 step:
digitalWrite(stepPin, HIGH);
delayMicroseconds(500);
digitalWrite(stepPin, LOW);
delayMicroseconds(500);
}
digitalWrite(8, HIGH); // non aktif

```

```

delay(1000);
digitalWrite(8, LOW); // aktif
digitalWrite(dirPin, LOW);
for (int i = 0; i < 3000; i++) {
    // These four lines result in 1 step:
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(500);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(500);
}
digitalWrite(8, HIGH); // non aktif
delay(1000);
}

void tanaman123() {
    Serial.println("Menyiram tanaman 1, 2, dan 3...");
    delay(2000);
    digitalWrite(8, LOW); // aktif
    digitalWrite(dirPin, HIGH);
    for (int i = 0; i < 1000; i++) {
        // These four lines result in 1 step:
        digitalWrite(stepPin, HIGH);
        delayMicroseconds(500);
        digitalWrite(stepPin, LOW);
        delayMicroseconds(500);
    }
    digitalWrite(8, HIGH); // non aktif
    delay(1000);
    digitalWrite(8, LOW); // aktif
    digitalWrite(dirPin, HIGH);
    for (int i = 0; i < 1000; i++) {
        // These four lines result in 1 step:
        digitalWrite(stepPin, HIGH);
        delayMicroseconds(500);
        digitalWrite(stepPin, LOW);
        delayMicroseconds(500);
    }
    digitalWrite(8, HIGH); // non aktif
    delay(1000);
    digitalWrite(8, LOW); // aktif
    digitalWrite(dirPin, HIGH);
    for (int i = 0; i < 1000; i++) {
        // These four lines result in 1 step:
        digitalWrite(stepPin, HIGH);
        delayMicroseconds(500);
    }
}

```

```

digitalWrite(stepPin, LOW);
delayMicroseconds(500);
}
digitalWrite(8, HIGH); // non aktif
delay(1000);
digitalWrite(8, LOW); // aktif
digitalWrite(dirPin, LOW);
for (int i = 0; i < 3000; i++) {
    // These four lines result in 1 step:
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(500);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(500);
}
digitalWrite(8, HIGH); // non aktif
delay(1000);
}

```

2. Dokumentasi

a. Bahan





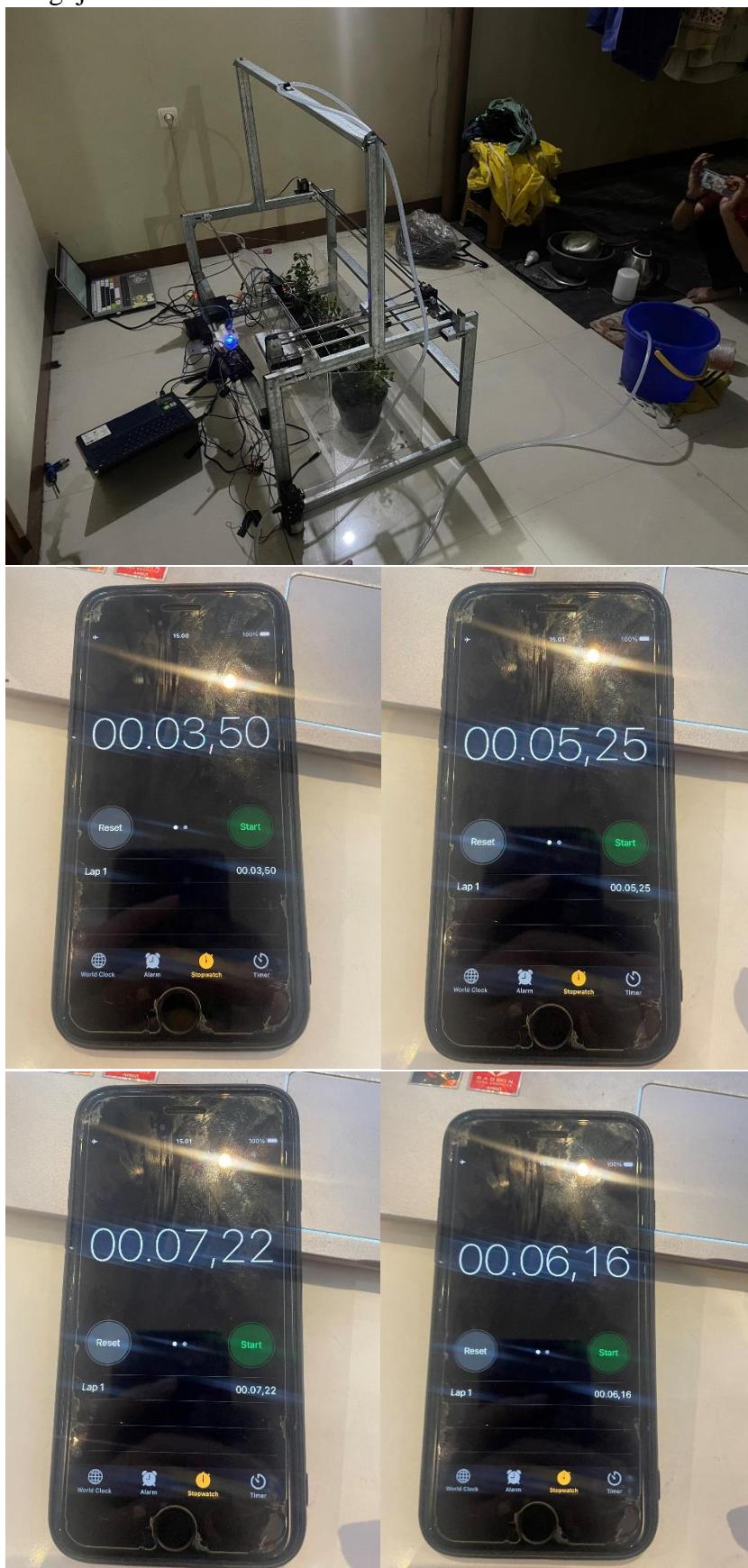


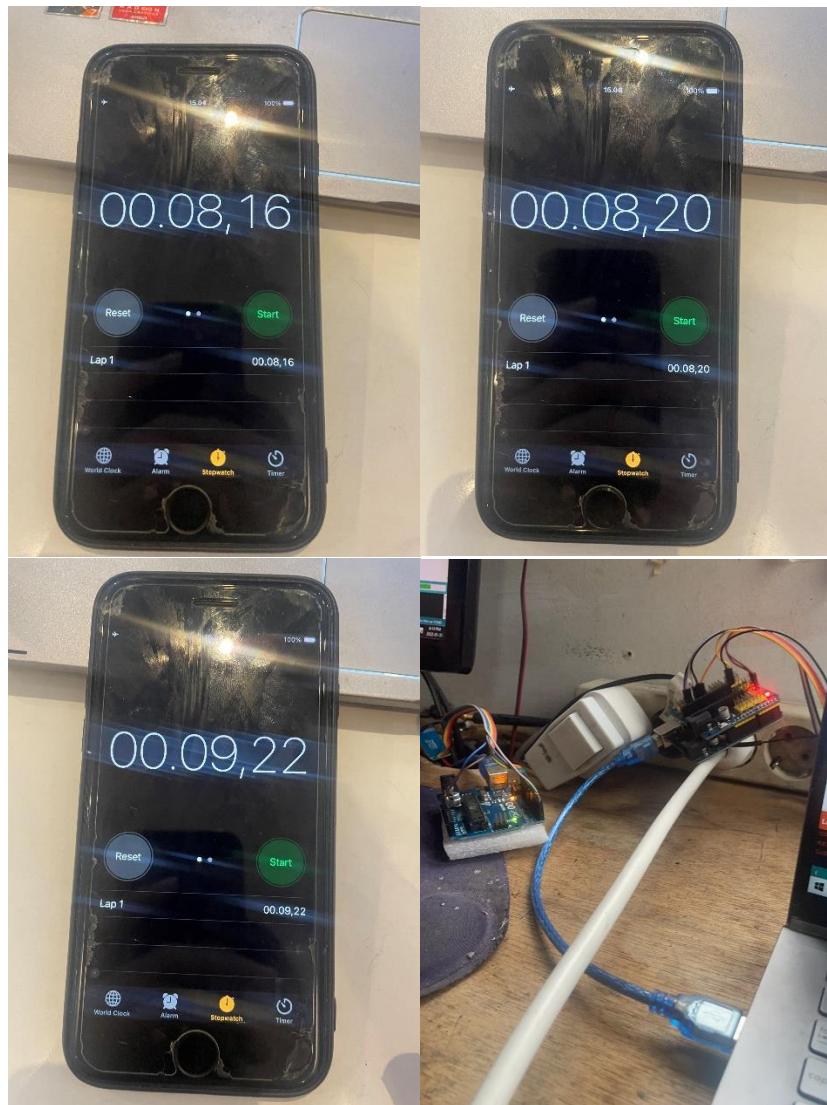
b. Pembangunan





c. Pengujian





3. Biaya Pembangunan Purwarupa

No	Benda	Harga Satuan	Unit	Biaya
1	Nema 17	Rp95.000/unit	2	Rp190.000
2	SC8UU	Rp22.500/unit	6	Rp135.000
3	GT2 Idler Pulley	Rp18.000/unit	2	Rp36.000
4	GT2 Pulley Timing	Rp12.000/unit	2	Rp24.000
5	Smooth Rod Steel 8 mm	Rp2.000/5 cm	56	Rp112.000
6	SFH8 8 mm	Rp15.000/unit	8	Rp120.000
7	Bracket L Socket	Rp31.500/unit	2	Rp63.000
8	Arduino Uno	Rp95.000/unit	1	Rp95.000
9	CNC Shield V3	Rp33.500/unit	1	Rp33.500
10	Timing Belt GT2	Rp13.500/meter	4	Rp54.000

11	DRV8825	Rp28.500/unit	2	Rp57.000
12	Besi Galvanum 4x2 cm	Rp120.000/5 m	2	Rp240.000
13	Baut, Mur, Ring	Rp15.000/paket	1	Rp15.000
14	Jasa Potong Besi	Rp35.000/jasa	1	Rp35.000
15	Pompa DC Mollar	Rp200.000/unit	1	Rp200.000
16	Nozzle Sprayer	Rp12.000/unit	1	Rp12.000
17	LoRa + Antena	Rp85.000/unit	1	Rp85.000
18	Relay, Transistor, Resistor	Rp15.000/paket	1	Rp15.000
19	Box X4 Custom	Rp50.000/unit	1	Rp50.000
TOTAL				Rp1.571.500